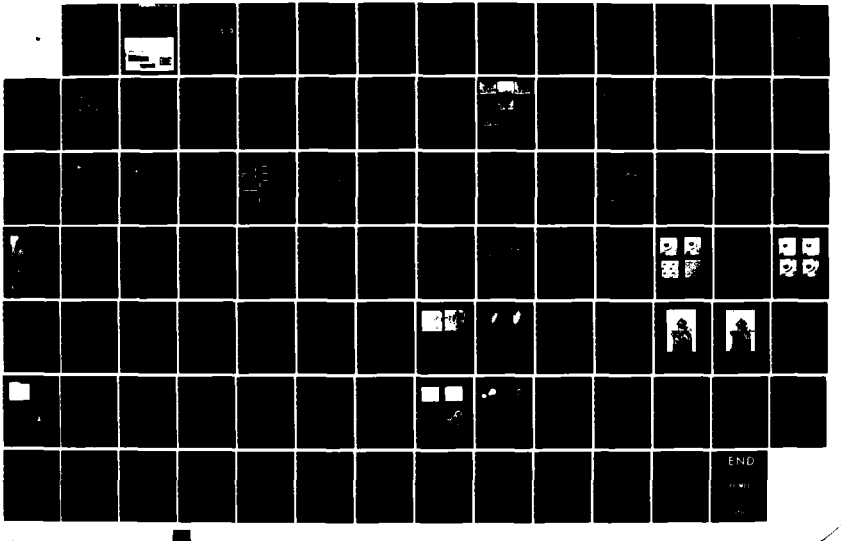
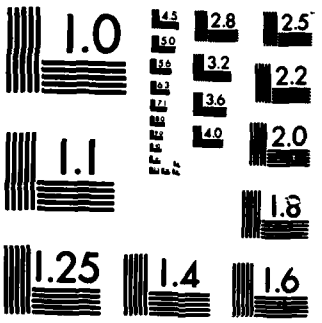
AD-A150 818 A SUMMARY OF IMAGE UNDERSTANDING RESEARCH AT THE UNIVERSITY OF MASSACHUSE..(U) MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INFORMATION S.. 1/1

UNCLASSIFIED A R HANSON ET AL. OCT 83 COINS-TR-83-35 F/G 9/2 NL





(2)

A SUMMARY OF IMAGE UNDERSTANDING RESEARCH AT THE UNIVERSITY OF MASSACHUSETTS*

Allen R. Hanson and Edward M. Riseman

COINS Technical Report 83-35

October 1983

Computer and Information Science

University of Massachusetts at Amherst

A SUMMARY OF IMAGE UNDERSTANDING RESEARCH AT THE UNIVERSITY OF MASSACHUSETTS*

Allen R. Hanson and Edward M. Riseman

COINS Technical Report 83-35

October 1983

Abstract

The major focus of our research program revolves around issues of static and dynamic image understanding. Our principle objective in this work is to confront fundamental problems in computer vision in the context of a large scale experimental system for interpretation of complex images. In this report we briefly review the current status of the VISIONS image understanding system, focussing on:

- the extraction of low-level syntactic descriptions of images,
- the representation of knowledge in a form suitable for use in the interpretation process,
- strategies for utilizing modular knowledge sources to link the sensory data to semantic hypotheses,
- inference mechanisms for integrating ambiguous and partial evidence from multiple sources, and
- control methodologies for both data-directed and knowledge-directed interpretation processes.

(continued on next page)

*This research was supported in part by the Defense Advanced Research Projects Agency under contract N00014-82-K-0464, by the Air Force Office of Scientific Research under contract F49620-83-C-0099, and by the Office of Naval Research under contract N00014-75-C-0459.

Our work in dynamic image interpretation (motion) is concerned with techniques for recovery of environmental information, such as depth maps of the visible surfaces, from a sequence of images produced by a sensor in motion. Algorithms that appear robust have been developed for constrained sensor motion such as pure translation, pure rotation, and motion constrained to a plane. Interesting algorithms with promising preliminary experimental results have also been developed for the case of general sensor motion in images where there are several significant depth discontinuities, and for scenes with multiple independently moving objects. A general hierarchical parallel algorithm for efficient feature matching has also been developed for applications in motion, stereo, and image registration. In addition, we have been designing a highly parallel architecture that integrates aspects of both parallel array processing and associative memories for real-time implementation of motion algorithms.

Table of Contents

0. Introduction

The work reported here outlines the current status of our research in static and dynamic computer vision. This research represents a continuing commitment to the long term goal of developing image interpretation systems for broad classes of complex scenes. Our work represents an experimental approach to the design and implementation of a large scale, complex system.

The goal of the VISIONS image understanding system is the transformation of a two-dimensional projection of a natural three-dimensional scene into a symbolic description of the world events comprising the scene, i.e., into a description which captures the meaning of the scene. Construction of the description, called an interpretation, involves identifying and representing the objects in the scene, their structure and relationships to each other, their approximate placement in three-dimensional space, and, quite possibly, their function and purpose. Such a description would permit the system to interact with and make predictions about its environment on the basis of visual data.

A methodological assumption underlying our work is that the interpretation process proceeds by making initial measurements on the image without knowledge of its contents. These measurements are then refined and associated with semantic events under the expectations and constraints provided by general knowledge of the physical world. This assumption implies that an abstract description of the image, in terms of measurable, primitive image events, must be obtained. The central research areas then include the extraction of syntactic descriptions of images, the representation of knowledge in a form suitable for use in the interpretation process, and the development of strategies for utilizing various sources of image and world knowledge to link the sensory data to semantic hypotheses.

One emphasis in the current work is on knowledge-directed interpretation via structures called schemas. One of the principles of the schema model of perception is that perception is guided by expectation. Perception of a scene is influenced and facilitated by expectations about the identity, properties, and relations among the objects being perceived. We have focussed on the representation and use of knowledge in the interpretation process, particularly the organization of knowledge about the world in such a way that links can be established between primitive image events and generalized semantic descriptions of those events.

A second aspect of our interpretation research involves examination of a range of control strategies for applying knowledge during the process of interpreting visual data. Issues include accessing relevant schemas based on prominent features, focus of attention mechanisms for selecting worthwhile portions of the sensory field for analysis, and ways of decomposing knowledge hierarchically so that partial matching can be effective. Control mechanisms should be able to exploit redundant knowledge and the physical constraints of the real world to reduce ambiguity.

Our work in dynamic image processing involves the investigation of several basic issues that must be understood in order to develop computer vision systems for terrestrial and airborne motion. From a sequence of images obtained from a sensor in motion, our goal is to demonstrate the feasibility of determining the changes in the sequence of images and establishing a consistent environmental model over time. The key scientific issue to be addressed is the recovery and effective representation of information concerning sensor motion, object motion, and the physical environment relative to the moving sensor. This would include the parameters of motion of the sensor and of any independently moving objects, as well as surface distance, extent, and orientation of the visible surfaces in

the environment. The necessary techniques are being developed using simulated and actual scenes with restricted forms of sensor motion, leading towards analysis of actual scenes with smooth, but arbitrary, motion. We have obtained extremely encouraging and robust results in the cases where sensor motion has been initially constrained to pure translation (i.e., linear motion with no rotation), pure rotation, and to motion constrained to a plane. These experimental results were obtained using image sequences of outdoor road scenes and industrial domains.

A second component of the motion research involves the efficient implementation of the basic procedures on massively parallel architectures, in this case the CAAPP Processor being developed within our architecture group. This effort is leading towards close to real time navigational systems. We also intend to integrate the environmental surface information into the VISIONS system.

I. The VISIONS Software Environment and System Tools

I.1. The Processing Cone

Given the long range goals of image understanding systems, one must consider the computational architectures that can facilitate the variety of forms of processing which most likely will be required. In almost any image analysis application, a characteristic which cannot be ignored is the massive amount of visual data which must be processed. For a full-color image of reasonable spatial resolution (512x512) and color resolution (3 colors, 6 bits/color), close to 5 million bits of information must be processed, often repeatedly. Faced with this computational overload, our group made a commitment to parallel processing at the very beginning of our research effort [HAN74, RIS74]. If such large amounts of sensory data are eventually to be processed by a machine in real time, then the use of large parallel array computers appears to be necessary. It is relevant to note that developments in technology imply that such devices could be economically feasible in the near future (see Section VI).

These considerations have led us to simulate a general, parallel computational structure, called a "processing cone", for manipulating large arrays of visual data. This parallel array computer is hierarchically organized into layers of decreasing spatial resolution so that information extracted from receptive fields of increasing sizes can be stored and further processed. The function of the processing cone is the transformation and reduction of the massive amount of image data in a form that facilitates scene interpretation by computer vision systems [HAN78b]. The hierarchy of computational processing provides a structure in which information at higher levels can direct more detailed processing at lower levels of the cone (for examples, see Section IV).

The processing cone is general-purpose in that it may be programmed by defining a prototype computation to be performed on a local window (i.e., subarray) of data. This prototype function is applied simultaneously — and in parallel -- to all local windows across the entire array. The user need only specify the definition of the function, the location of the source(s) of the data within the cone, a description of the size and shape of the local window, and the destination of the result(s) within the cone (Figure 1). The cone's operating system simulates lockstep computation just as if there were parallel arrays of synchronous microprocessors computing on each window; each microprocessor executes a copy of the prototype computation.

There are three basic modes of procesing available within the cone: reduction operations, horizontal (or lateral) operations, and projection operations. These correspond to a flow of information up, laterally, and down the cone, respectively, as shown in Figure 1. During a reduction process upward through the layers of the cone, a window of data at level k is processed and the resultant value(s) stored at level k-1; the data is reduced since the central portions of each window are non-overlapping. During horizontal operations, the domain and range of the local function are the same level of the cone, which means that processing is restricted to a single level. The resolution of the data remains constant since each cell at a level will have a window centered over it. This same cell receives the result of the local applications of the function, but note that each cell can receive and store a vector of values.

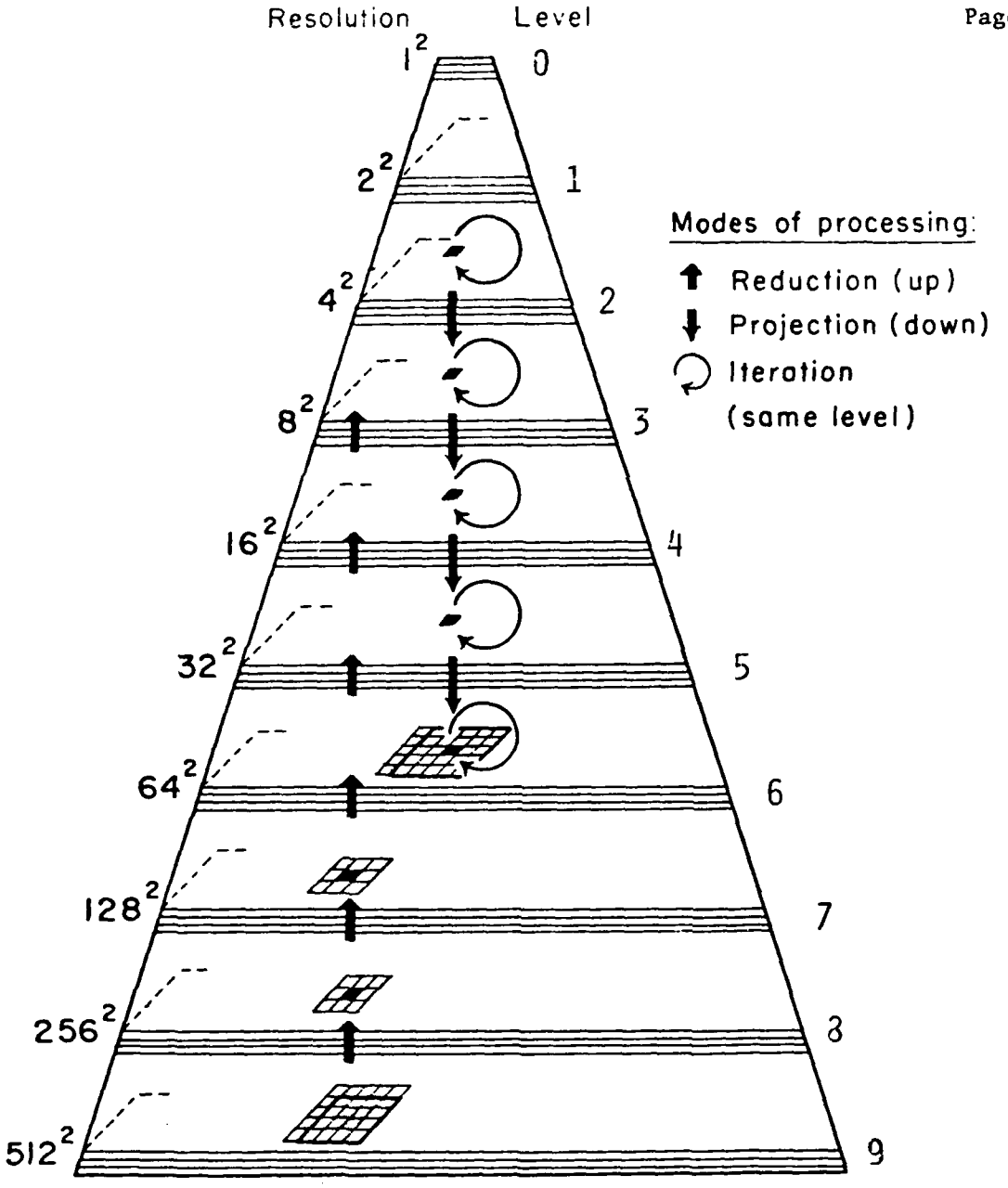
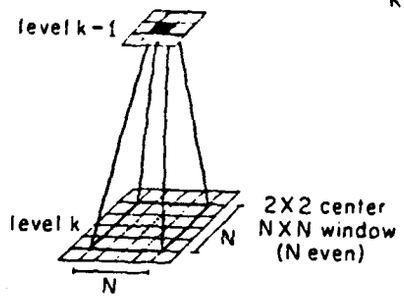


Figure 1(a).

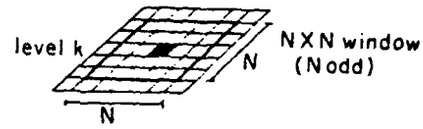
The processing cone is a parallel array computer hierarchically organized into layers of decreasing spatial resolution. Information within the cone is transformed by means of functions operating on local windows of data. The results of the function are stored in one or more "planes" of data at specified levels. Cone algorithms are specified as sequences of these parallel functions applied in one of three processing modes: reduction (processing up the cone), projection (processing down the cone), and iteration (processing at the same level).

(i) Reduction processing: $f^{t}_{k,k-1}(I_1,\ldots,I_D;O_1,\ O_R)$



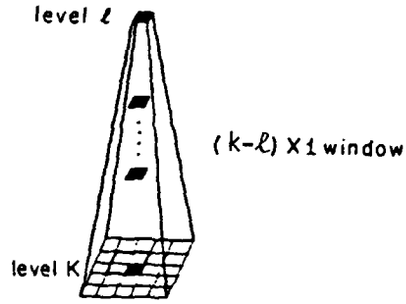
Every cell at level k-1 is associated with a unique 2X2 window of cells at level k.

(ii) Horizontal processing: $f^{t}_{k,k}(I_1,\ldots,I_D;O_1,\ldots O_R)$



Every cell at level k is treated as the central cell of an NXN window of cells at level k.

(iii) Projection processing: $f^{t}_{(k-1,k-2,\ldots,\ell+1,\ell;k)}(I_1,\ldots,I_D;\ O_1,\ldots O_R)$



Every cell at level k is associated with a window of ancestral cells from level k-1 through the top of the cone, one from each level. The particular cell is determined by the sequence of reduction windows.

Figure 1(b).

Processing modes in the cone. (i) During reduction processing, the local function f is applied in parallel to all even-sized windows of data at level k (the input data is in planes $I_1,\ldots,I_D$). Results are stored in the output planes $O_1,\ldots,O_R$ at level k-1. (ii) During horizontal (or iterative) processing, the input data for f is derived from odd-sized windows and the results are stored at the same level in the cone. (iii) During projection, the input data for f is obtained from levels higher in the cone. Results are stored in the output planes at level k.

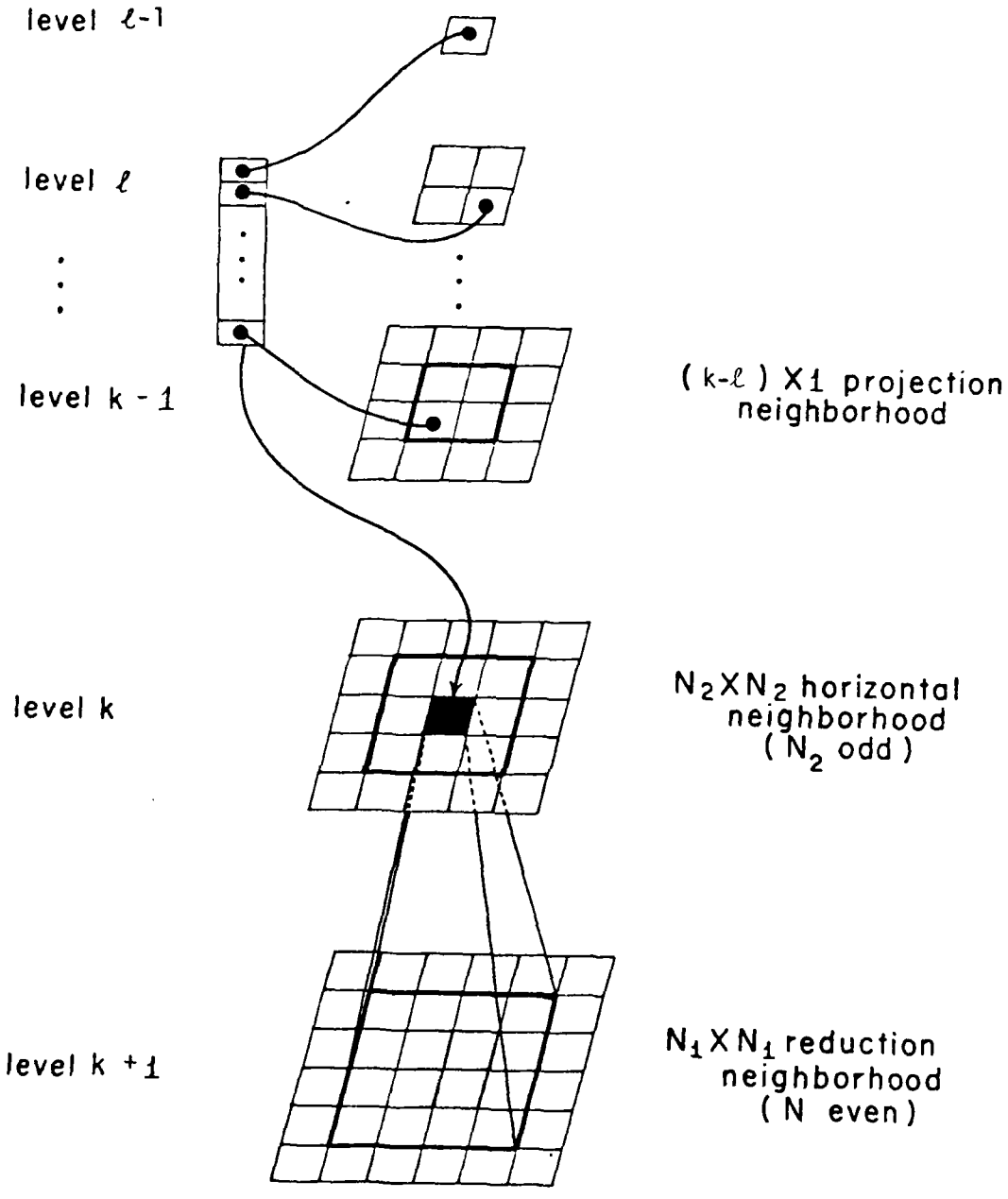


Figure 1(c).

The full local neighborhood of a cell in the cone. Given a particular cell at level k, the value placed there can be computed from the data which is simultaneously available from the neighborhoods defined for the three processing modes. Thus, the domain of the local function is the union of the three types of windows associated with the three pure processing modes defined in (b).

During projection operations, information in upper layers of the cone influence computation at lower layers. This is achieved by extending the definition of the neighborhood for horizontal processing to include data present in parent cells in the hierarchy above.

The full neighborhood of a cell at level k is shown in Figure 1(c). If the reduction neighborhood is 4x4 and the horizontal neighborhood is 5x5, then a cell at level k will have simultaneously available the storage planes of 16 cells at level k+1 (the reduction neighborhood), 25 cells at level k (the horizontal neighborhood) and the unique set of k ancestral cells from levels k-1 through level 0 (the apex of the cone).

I.2. Image Operating System

In order to carry out complex image interpretation experiments of the type described in subsequent sections, a development environment which would flexibly support such experimentation was needed. In response to this need, we have developed an extensive, interactive software environment called the Image Operating System [KOH82, KOH83] and have maintained a long range commitment to its continued evolution.

The VISIONS researcher has access to an image operating system (IOS) which consists of a high-level interpretive control language (LISP) supporting efficient "image operators" in a non-interpretive language (Figure 2). This environment is based on the processing cone discussed in the previous section. The IOS, implemented on a VAX 11/780 running VMS, is a powerful tool which has been used to carry out all low-level image analysis research for the VISIONS image interpretation project since 1979. Features of the IOS include:

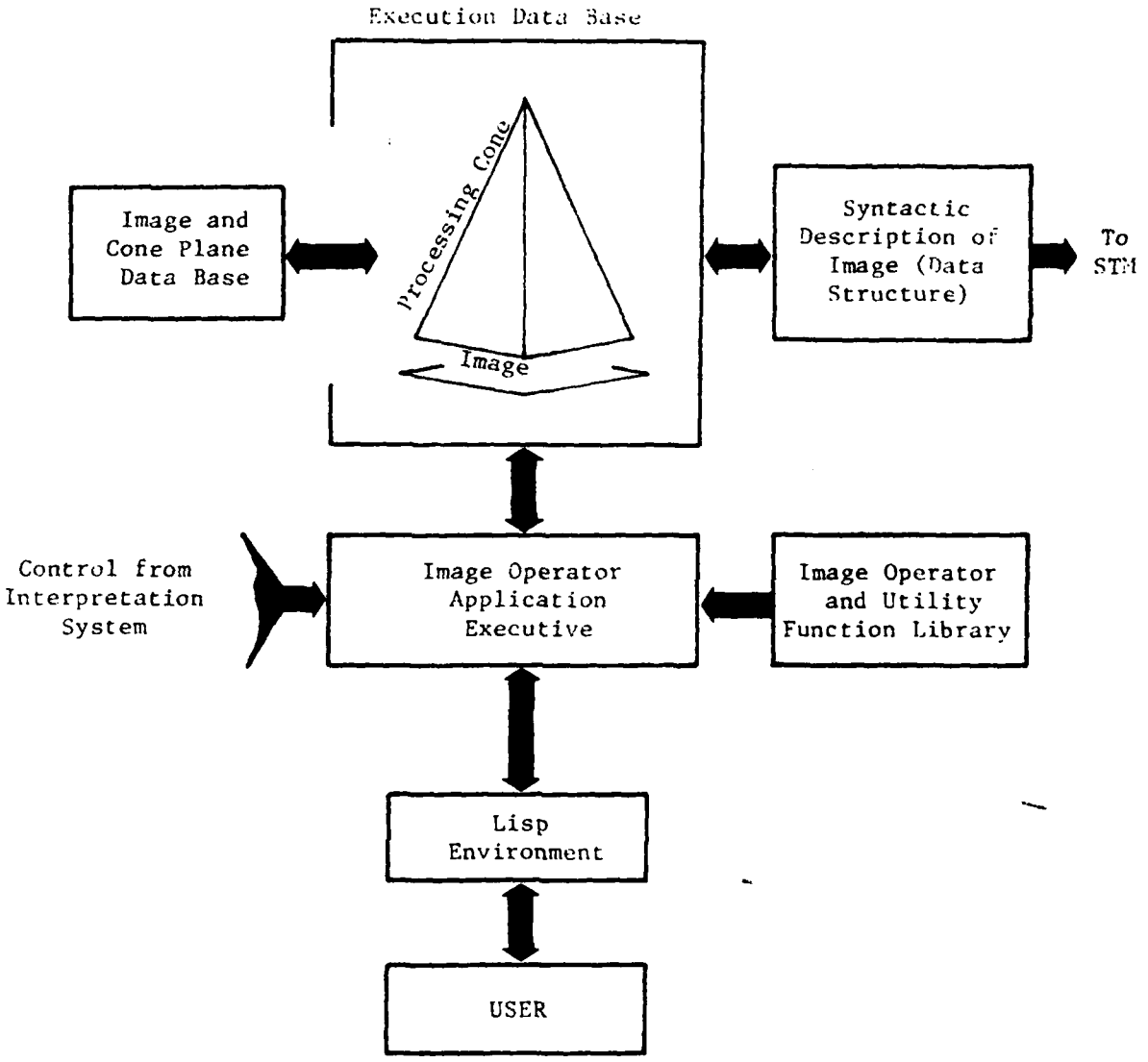


Figure 2. Block Diagram of the Image Operating System.

The VISIONS image operating system is a sophisticated software environment for developing and evaluating image segmentation algorithms. Built around the hierarchical processing cone, it consists of a high-level interpretive control language and efficient image operators written in a noninterpretive language. The image operators are viewed as local operators applied in parallel to all pixels in the input image(s). Complexes of image operators are coordinated from the LISP environment. The resulting syntactic descriptions are represented in short term memory (STM) in a structure called the region, segments, and vertices graph (RSV graph); this data forms the basis for the interpretation processes. Specific requests for further processing are fed back from the interpretation system and the RSV structure is updated accordingly.

1. image data base and disk filing capabilities,

2. the data structures necessary to implement a generalized hierarchical cone structure,

3. mechanisms for defining parallel prototypical functions (image operators) expressed as Fortran programs,

4. methods for applying image operators,

5. methods for specifying variable and plane bindings for image operator parameters and logical planes,

6. interactive mechanisms for composing sets of image operators into complex algorithms via LISP invocation,

7. highly flexible color graphics display capabilities,

8. an error handling system which assists the user to dynamically correct error conditions,

9. a user help system for handling default conditions and describing function parameter specification,

10. automatic documentation system for maintaining processing history with image data.

LISP was adopted as the language for interfacing the Image Operating System with the user. This was a natural choice for a number of reasons:

1. LISP provides powerful control structure capabilities.

2. LISP is interpretive and therefore permits the dynamic generation of experiments which would otherwise require non-interactive compilation and link steps in a non-interpretive language.

3. LISP would provide a uniform interface with the semantic interpretation component of the VISIONS system, since the interpretation system has been developed in LISP and a graph processing language called GRASPER [LOW78], which is, in turn, built on LISP.

The use of LISP for the entire implementation has the desirable quality of unifying the system implementation and making the portability of the system much more feasible. However, a non-interpretive language was chosen for implementation of the underlying Image Operating System and the image operators (FORTRAN and VAX-MACRO) for reasons of efficiency. Thus, LISP provides flexible user control over the remainder of the underlying system. Within the uniform structure of the IOS, a wide variety of experimental tools and image functions have been implemented [KOH83].

I.3. GRASPER Extensions to LISP

The high level interpretation processes of the VISIONS system are implemented in GRASPER [LOW78, LOW79, WIL77], a graph processing extension to LISP. GRASPER supports large, dynamic graph structures consisting of nodes, edges (directed connections between pairs of nodes), and spaces; each of these primitives have names and values. Spaces are subsets of nodes, edges, and values (i.e., subgraphs). GRASPER graphs are easily created, queried, modified, and deleted through a small set of operators which may be composed into larger operators. All GRASPER operators are applied to virtual spaces which are defined as the union of a set of other spaces. GRASPER also supports a virtual memory system for graphs; large graphs are partitioned into pages and moved between primary and secondary memory as required.

GRASPER provides a uniform, natural structure for the implementation of the knowledge structures in VISIONS. The virtual space structure supports the hierarchical organization of both long-term and short-term memory in the VISIONS system and allows specific knowledge networks to be dynamically constructed from subnetworks. This facility supports a focus of attention mechanism under the

guidance of schema directed processing (Section III.3), and the common LISP substrate of the IOS and GRASPER provides the interface between the components of the system. Over the last year, we have been implementing in this combined system:

a) new segmentation and feature extraction routines;

b) an interface between the segmentation system and the interpretation system;

c) a knowledge base of modular object hypothesis rules;

d) schema-based interpretation and control mechanisms;

e) an inference engine to propagate the effects of partial evidence;

f) extensions to long- and short-term memory.

II. Segmentation

Algorithms designed to be incorporated into an image understanding system must extract a variety of information [HAN78c] from a scene and often can become quite complex. One class of problems involves the segmentation of an image, which is the partitioning of an image into areas -- or regions -- based on invariance of some subset of visual features. A segmentation of an image is a partition of the picture elements (or pixels) into disjoint sets (regions) of spatially contiguous pixels. The goal of the image segmentation algorithms is to produce segmentations for which there is a high correlation between the entities of the real world (objects, parts of objects, and surfaces) and the regions of the segmentation.

It is difficult to overstate the complexity of the segmentation problem. In natural, unconstrained scenes, such as full color outdoor scenes, any straight-forward approach is prone to gross errors. Inherent difficulties of the scene such as direct and indirect lighting, varying orientation of surfaces, shadows, texture, specularity, and noise in the segmentation system (especially due to the discrete digital representation) make the generation of "good" segmentations very difficult. If the image being analyzed has any significant degree of textural variation, then the problems encountered in extracting this information are greatly magnified. In such cases it is necessary to extract features that typify the textural variation in order to carry out the segmentation.

Due to the problems outlined above, the goal of image segmentation has become controversial and is viewed by many as an ill-formed task which is dependent upon the goals of the processing and the domain being analyzed. Some have chosen as a goal the recovery of intrinsic image properties which are a function of the physical environment and can be precisely specified, e.g., surface properties of depth, orientation, and reflectance. Nevertheless, the presence of some form of segmentation will be unavoidable since the reliable extraction of surface properties in complex domains has not yet been achieved and remains an extremely difficult problem. Certainly all interpretation systems must extract a set of image features upon which the higher level processing is based.

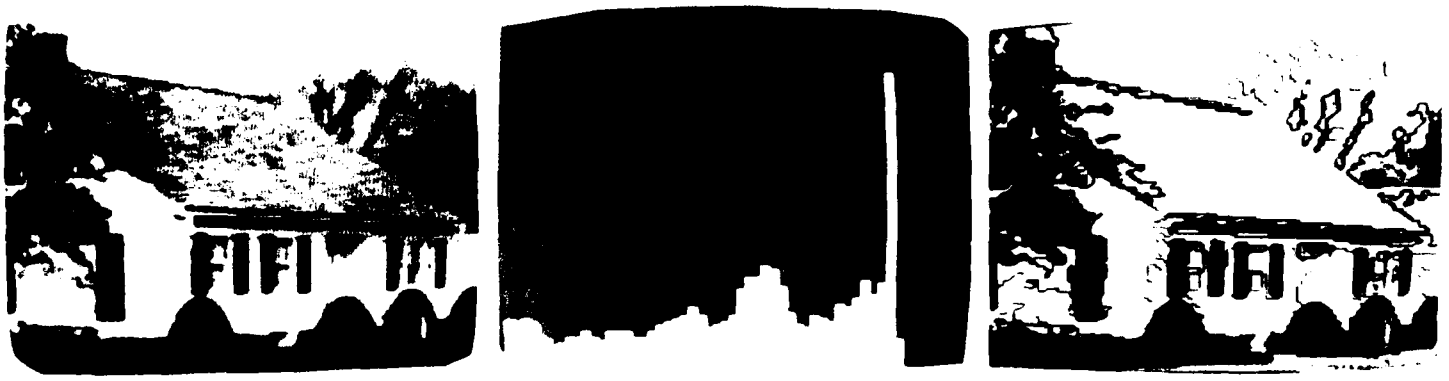
Thus, the problem of image interpretation is often split into two stages. The first stage segments the scene while the second stage attempts to label objects and build a three-dimensional model of the scene using the segmentation, the image, and prior world knowledge. In order to deal with the inevitable errors in initial segmentation, we have closed the feedback loop and performed resegmentation under semantic guidance (see Section III.2 and Figure 2).

Past efforts of our group in image segmentation are documented in [HAN78a, NAG79, PRA80, KOH81, HAN80a, OVE79]. Recent work in image segmentation has involved the improvement of our general region segmentation algorithms, the extraction of straight lines of both low and high contrast, and the development of a uniform segmentation system to act as an intermediary between the interpretation system and the segmentation/feature extraction processes.

II.1. Region Segmentation

Algorithms for region formation usually take advantage of the similarity of pixel feature values, rather than discontinuities in feature values as in the case of edge/boundary algorithms. We have developed a family of region-formation algorithms which utilize peaks of activity, or "clusters", in feature space to form pixel labels, which can then be grouped into region labels via connected component analysis [NAG79, NAG81a,b,c, KOH81, KOH83].

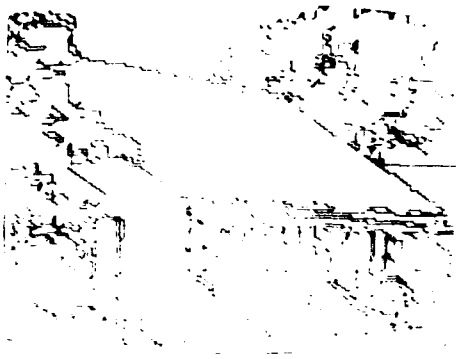
In these algorithms, prominent peaks in a one- or two-dimensional histogram of feature values are used to assign a vector of region labels and associated confidences to each pixel in the image. The confidences are computed as a function of the distance in feature space between the pixel feature value(s) and the cluster centers (the peaks). Nagin [NAG79, NAG81b] follows this step with a relaxation labelling process which updates the confidences associated with the region labels based upon their compatibility with neighboring pixels to arrive at a "consistent" labelling (Figure 3). These relaxation labelling algorithms are computationally expensive due to the iterative nature of the computations performed at each pixel. Kohler [KOH83] has recently shown that only a small fraction of the pixels actually benefit from the iterative update in the sense that the maximum likelihood labels at each pixel (based on the confidence values) would be modified by this contextual updating process. The final results appear to be much more dependent upon the proper selection of the peaks in the feature histogram.



(a) (b) (c)

(d)

(e)

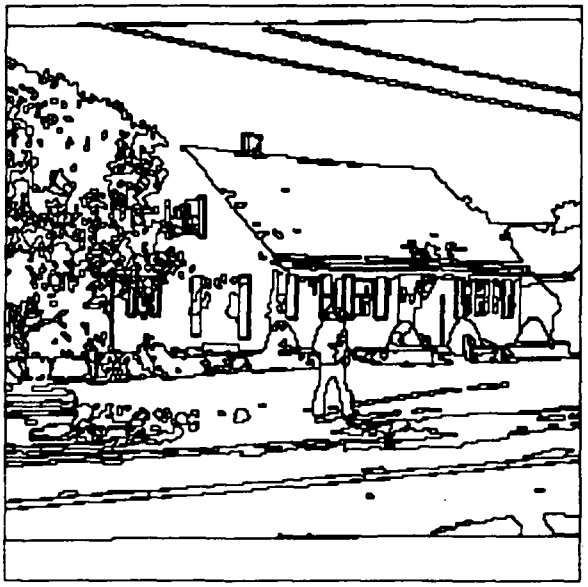


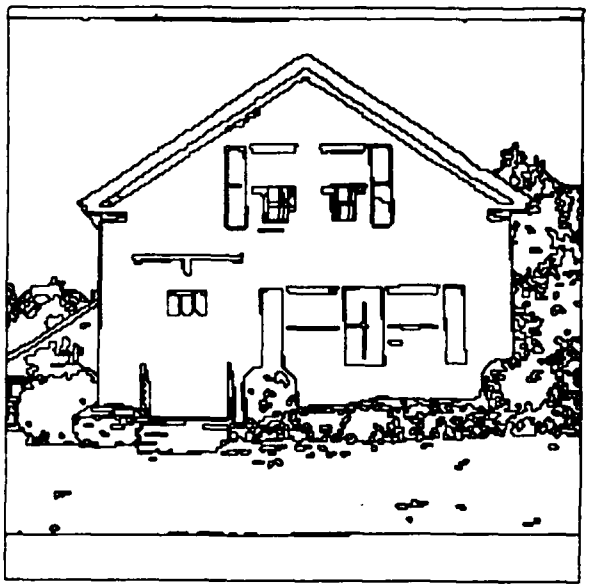
(f)

Figure 3. Results from the Nagin Segmentation Algorithm. (a) Original image. (b) Global histogram of an image feature. Cluster centers are determined and a probabilistic region labelling is formed in which each pixel has a vector of possible region labels and associated likelihoods. (c) Segmentation resulting from probabilistic labeling after a relaxation algorithm was used to update pixel label likelihoods on the basis of local spatial information. Note that some detail is lost, particularly along the boundary between the roof and garage. (d) By dividing the image into local sectors and applying the process in each sector, local events which may have been lost in the global view may be retained. (e) Results of localized algorithm after merging across the artificial sector boundaries; the results are displayed as edges superimposed over the original image. (f) Same as (e) but showing only the region boundaries; note that more image detail is preserved (compare with (c)).

One problem commonly encountered with histogram clustering algorithms is that local structure or fine structure is often masked by the global nature of the histograms; for example small but important peaks of feature activity may be hidden by larger clusters of activity, possibly in remote portions of the image. It is obvious that large and/or distant regions should not interface with the extraction of locally distinct regions. Nagin developed methods for partially overcoming these problems by subdividing the image into regular overlapping subimages, with the segmentation process being applied independently to each subimage. The artificial boundaries produced by the subdivision process are eliminated by merging regions across these boundaries on the basis of the similarity of region features. Kohler [KOH83] developed a cluster addition process which proposes candidate clusters from adjacent subimages for which only marginal evidence exists in the feature distribution of the central subimage. This process substantially improves the sensitivity of the segmentation process to weak clusters, and makes the merging process in the last step more reliable (Figure 4).

The observations in the previous discussion have recently led to the development (not yet reported) of a region algorithm based on local merging of conservatively formed regions. The merging algorithm is independent of the techniques used to form the initial segmentation; the only requirement is that a conservative segmentation be produced, since only the merging of regions (as opposed to the splitting of regions) will take place. Thus, the initial segmentation should be highly overfragmented so that it is composed of very small and mostly uniform regions. The algorithm first merges very small one and two pixel regions into surrounding regions. The remaining regions are repeatedly merged using a set of merge rules expressed as LISP functions. Each merge rule may or may not be applicable to a given merge decision, and if

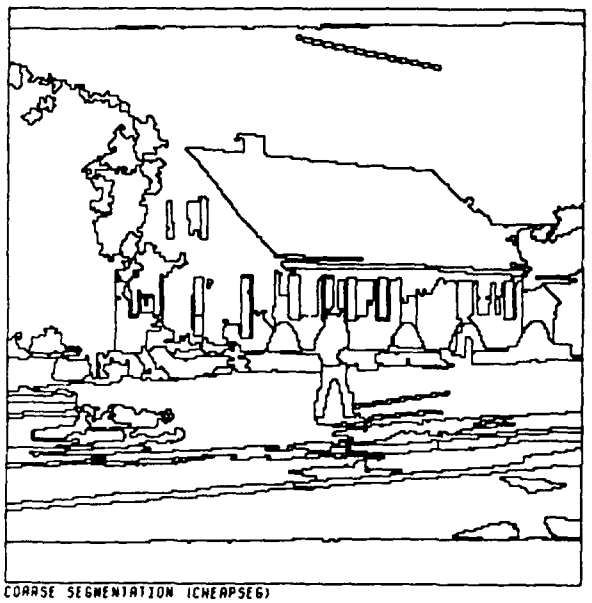


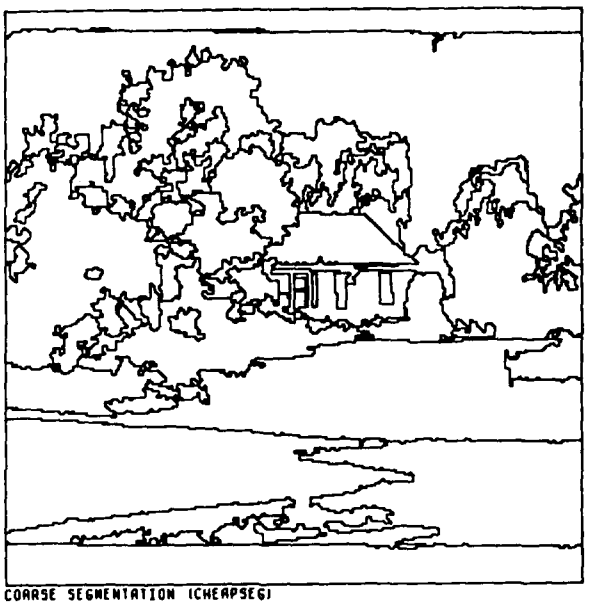


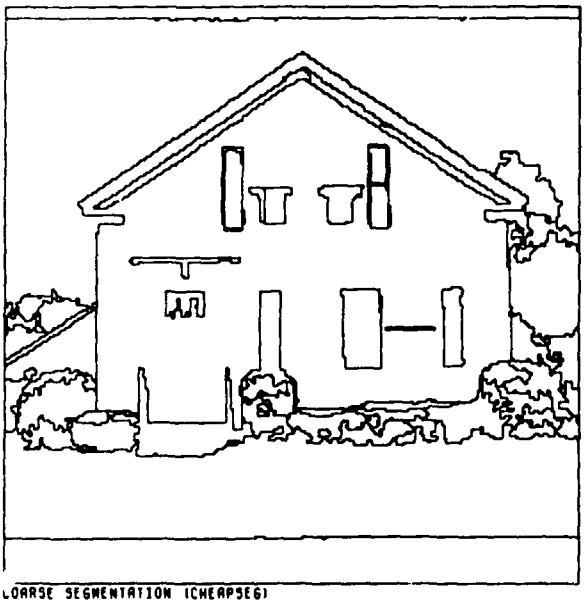
(a)

Figure 4. Sample Segmentations.

(a) Results from the Nagin-Kohler algorithm. The image is first subdivided into rectangular sectors (32×32 or 64×64) which overlap neighboring sectors by a few pixels. Within each sector, a feature histogram (here intensity) is formed and a peak/valley analysis performed. Each prominent peak defines a cluster center and the valleys on either side define the extent of the cluster. The set of clusters found is modified to include similar clusters from surrounding sectors. Based on the augmented cluster set, a region labelling is performed. Regions are then merged across the artificial sector boundaries on the basis of region statistics adjacent to the sector boundaries, and small (1 or 2 pixel) regions are merged into surrounding regions. A multi-pass region merging algorithm is then applied which merges across region boundaries using global region statistics. The algorithm terminates when no further merges take place.







(b)

Figure 4, continued.

Sample Segmentations.
(b) Results from a modified Nagin-Kohler algorithm. This version of the algorithm was modified for computational efficiency. The image is subdivided into sectors and the peak/valley analysis is performed but without the peak addition step. Regions are merged across the sector boundaries and then small regions are merged into surrounding regions. A single pass region merging algorithm is then applied which merges across region boundaries on the basis of region statistics weighted towards a merge as an inverse function of region size. The resulting segmentations have a tendency to be somewhat coarser than those produced by the more computationally expensive version of the algorithm; compare with Figure 4a.

applicable the rule contributes a weighted vote either in favor of the merge or opposed to the merge. The weight of the vote is proportional to the power of the rule and the confidence of the merge/nomerge decision. Rules utilize local and global region feature means, variances, size, adjacency, and gradient characteristics.

The effectiveness of this approach lies in the ability to add explicit rules to enforce specific types of merges. These rules can be modified and controlled by knowledge of specific characteristics of the regions desired (e.g., under direction of the interpretation system). The approach seems to be remarkably robust on a wide variety of imagery, including biomedical, outdoor, and remotely-sensed earth images. It preserves finer structure and is more sensitive to local differences across boundaries (Figure 5). The disadvantage of this approach is the very large number of regions that must be used initially (over-fragmentation) to guarantee effective results. It may be possible to employ a version of the Nagin-Kohler local histogram algorithm, set to extract a larger number of clusters, as the initial mechanism for producing an intermediate number of regions as input to the rule-based merging algorithm.

II.2. A New Approach to Extracting Linear Features

Recently, Burns [BUR83] has been investigating techniques for the extraction of straight lines from images. In many cases the presence of long straight lines in an image indicate regular geometric structures; for example the boundaries of houses, roads, and many human artifacts are bounded by straight lines. Such areas are usually semantically important and deserving of some attention from the interpretation process. Short lines can provide very useful texture properties. The problem of reliably extracting straight lines

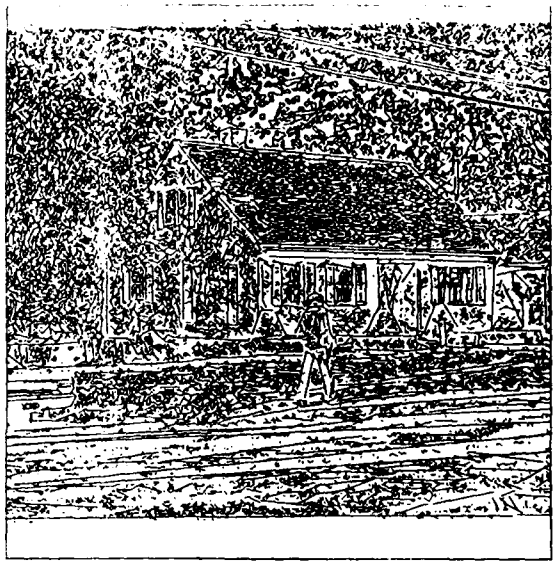
Figure 5. Sample Segmentations from the Rule-Based Segmentation Algorithm.

This algorithm is based on an oversegmentation and merging strategy. It assumes as input a very conservatively formed region segmentation consisting of a large number of regions which are almost uniform in a set of features. A rule-based multi-pass merging algorithm is then applied until no further merges are performed. The rule set is variable and can be made context-sensitive in the sense that high level goals and/or results from a partial interpretation can be used to modify an existing rule set or to select an alternate rule set. For the segmentations shown here, the rule set included rules reflecting global region statistics, length of the common boundary, several texture measures, anti-aliasing, etc.

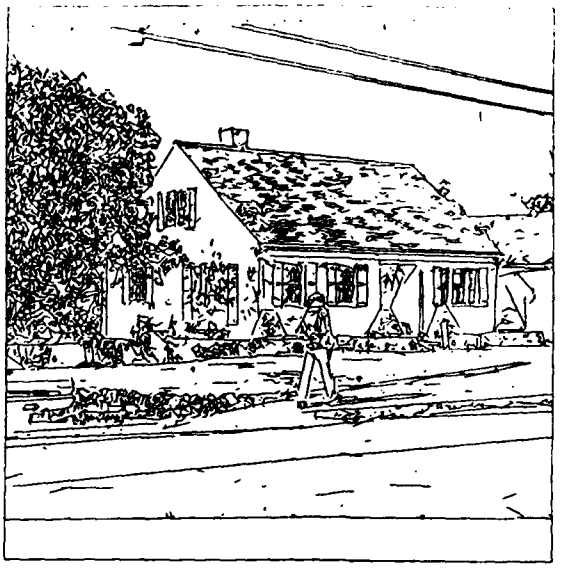
has remained surprisingly difficult; most algorithms in the literature produce fragmented and missing lines even in cases where the presence of these lines seems very clear to human observers.

Burns' goal is to detect and represent only straight lines in an image and to accurately measure the position of the line and its orientation. The general approach involves computing local estimates of gradient magnitude and direction, grouping pixels associated with lines, and then using these regions to extract the line. The unique aspect of the algorithm is that the grouping process relies entirely on the orientation of the gradient while the magnitude of the gradient is only used for placement of the line. Thus, a gradient orientation segmentation produces regions of pixels with roughly uniform gradient. The line is extracted by fitting a plane to the gradient magnitude and then intersecting this plane with the mean of the pixel values weighted by gradient magnitude. Endpoints are determined from the extent of the sample region. The resulting lines are parameterized and filtered as a function of length, contrast, and sharpness.

The results from this algorithm are extremely promising (Figure 6). We are examining ways in which the partitioned line sets can be used to determine vanishing points and surface orientation, measure simple shape features, measure surface texture, determine slow gradients, and extract shading information. The line sets appear to be useful for merging regions on the basis of similar line features. Short lines of similar contrast and orientation can be used for grouping pixels associated with textured regions such as shingled roofs. Tree foliage, on the other hand, seems to produce short lines of highly varying orientation. We are also looking at appropriate parameterization of the resulting lines and at the extension of this technique to curve extraction (e.g., quadratic arcs).

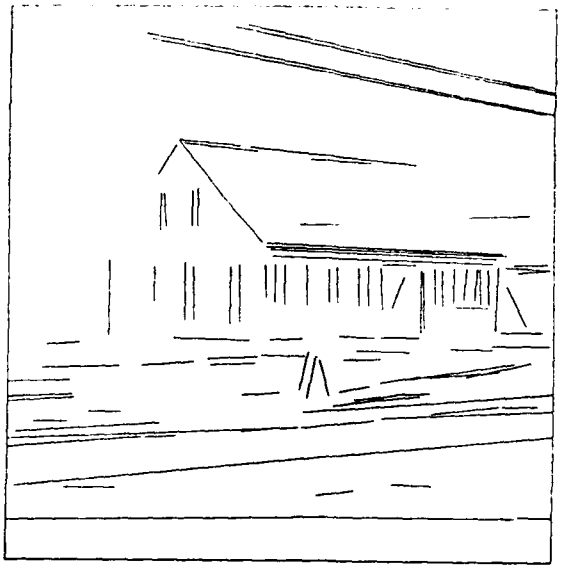


(a)



(b)

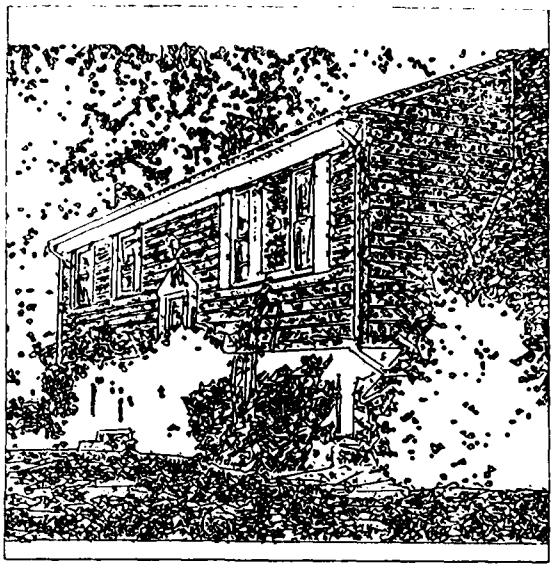
(c)



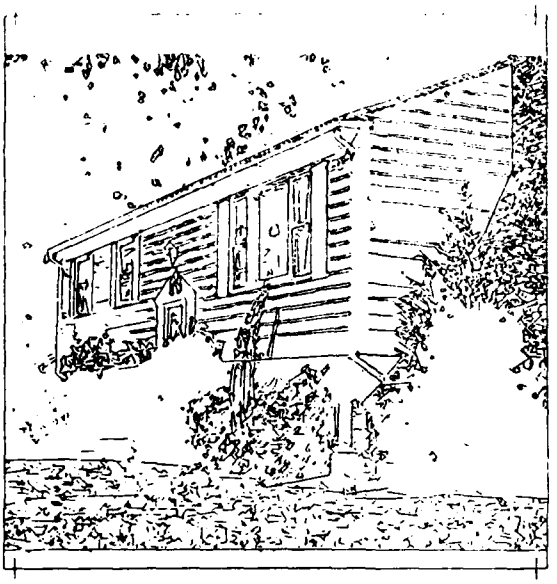
(d)

Figure 6. Results from the Linear Feature Extraction Algorithm.

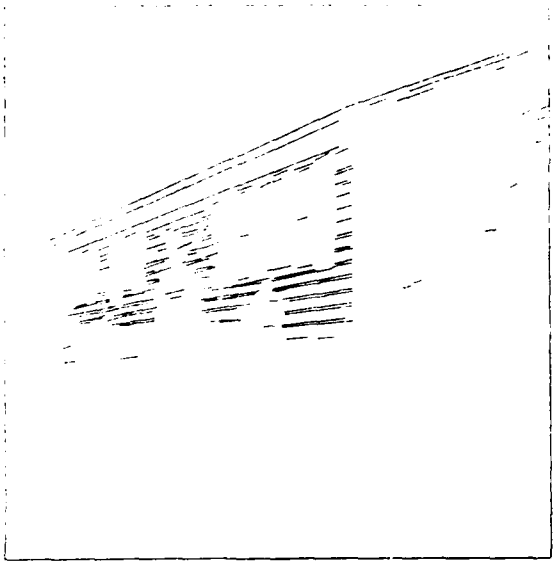
(a) All lines found by the algorithm. (b) Results of filtering (a) on the basis of gradient steepness (≥ 10 gray levels per pixel). (c) Filtering (a) for short, high contrast edges produces the basis for texture descriptors. (d) Filtering (a) for long, high contrast lines results in many of the visually meaningful structural edges.



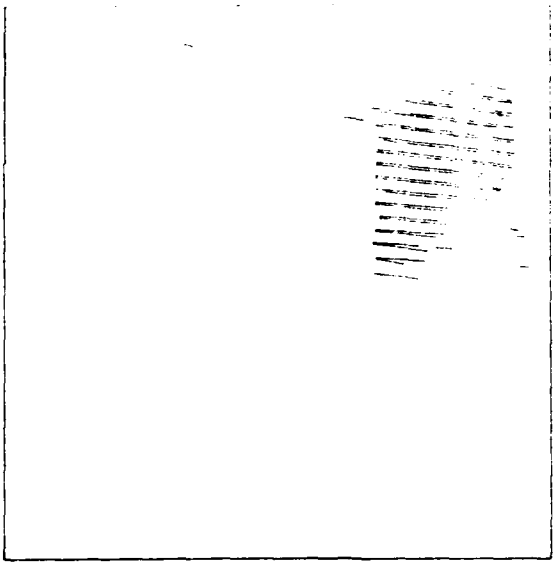
(e)



(f)



(g)



(h)

Figure 6, continued.

(e) Initial set of lines. (f) Results of filtering (e) on gradient steepness (≥ 10 gray levels/pixel). (g-h) Results of filtering (e) on orientation and spatial location. (g) Orientation in the range 3-28°. (h) Orientation in the range 165-177°. The orientation filtering produces results which could be used for perspective analyses.

III. Image Interpretation

III.1. Introduction

The VISIONS Image Understanding System is an experimental testbed for examining issues in knowledge directed processing and the construction of integrated computer vision systems [HAN78a,b]; see Figure 7. The goal is to provide an analysis of color images of outdoor scenes, from segmentation through the final stages of symbolic interpretation of that image. The output of the system is intended to be a symbolic representation of the three-dimensional world depicted in the two-dimensional image. This involves the determination of object labels for major image regions and an approximate placement of objects in three-dimensional space, which then allows the system to predict from this representation the rough appearance of the scene from other points of view.

The general goal of the segmentation processes in VISIONS is to provide a syntactic description of the image, which includes the extraction of primitive elements of the image (such as regions, edges, lines, and surfaces) and selected features of these primitive elements (such as color, texture, shape, orientation, etc.); see Figure 8. In VISIONS, a segmentation executive builds an initial segmentation of the scene, which is then used by the image interpretation system to build a set of hierarchically structured hypotheses about the particular scene based on stored world knowledge. When necessary, these hypotheses about the semantic content of the scene can be used to produce feedback requests to the segmentation executive to modify or refine the segmentation. This implies that the initial segmentation need not be "ideal", but it must be sufficiently detailed to allow the interpretation system to extract general image properties in order to begin goal-directed processing.

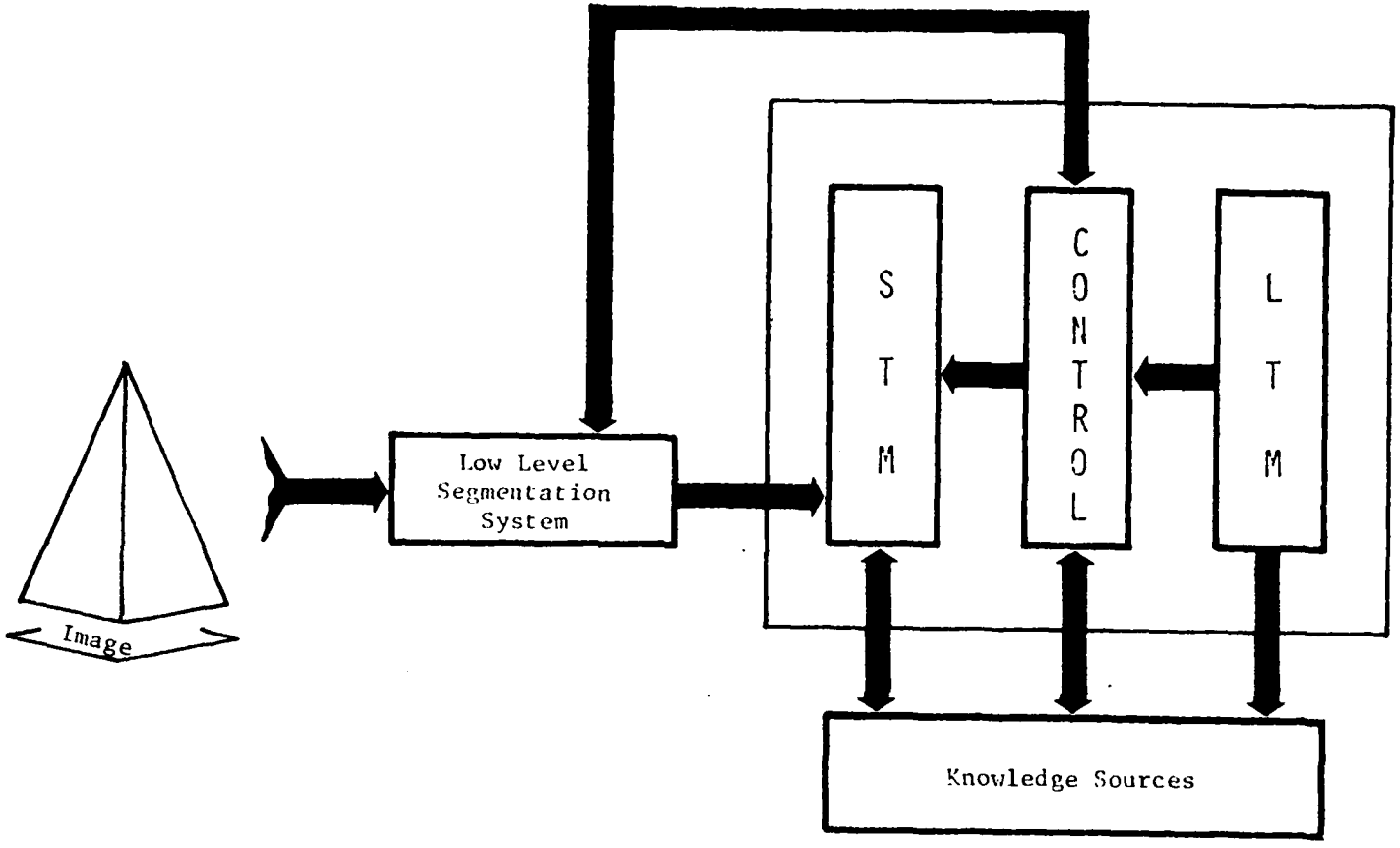


Figure 7. Block Diagram of VISIONS.

The low level segmentation system provides a syntactic description of the image to be interpreted. This description is stored in short term memory (STM), where an interpretation is incrementally constructed via application of modular knowledge sources (KSs) operating under the constraints of world knowledge stored in long term memory (LTM).

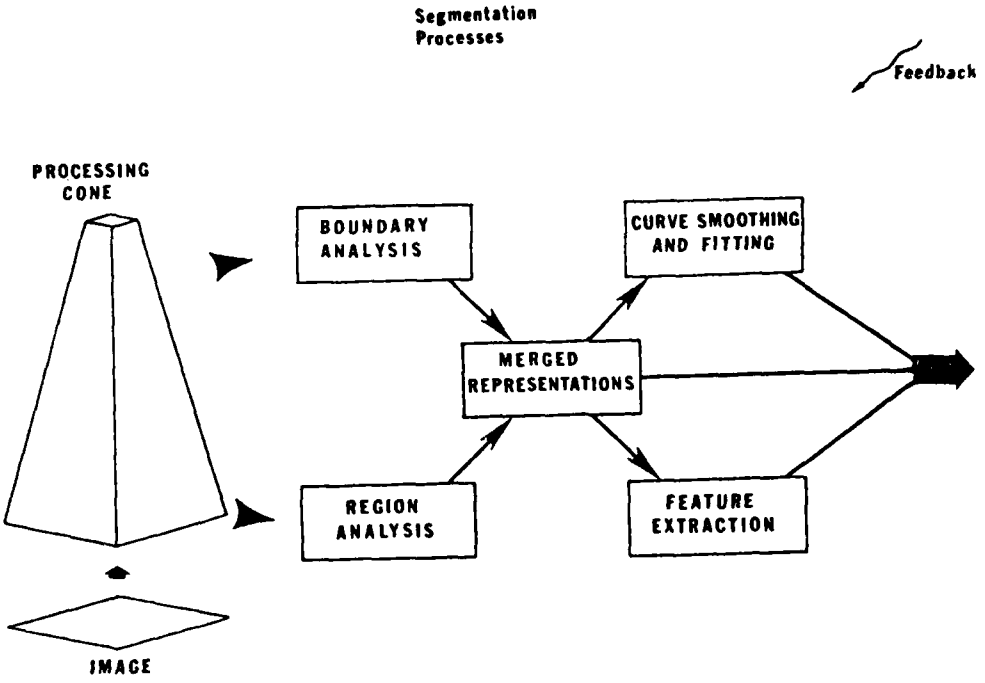


Figure 8. Overview of the Segmentation Processes in VISIONS.

Region and boundary analysis algorithms are implemented in the image operating system as complexes of user functions. The resulting segmentations can be merged and attributes of the regions and lines extracted. The result of this processing is a layered graph representation of the syntactic structure of the image which forms the basis for subsequent interpretation processes. The segmentation processes can be made sensitive to the context provided by partial interpretations via specific requests fed back from the interpretation processes.

An interpretation is created by grouping the visual primitives in the appropriate ways and linking them to semantic labels under the constraints imposed by world knowledge contained in long term memory (LTM). This process is accomplished by applying sequences of knowledge sources (KSs) which are modular processes governing the transformation of data between particular levels of representation. The KS application takes place under the guidance of a control strategy, and extends a partially constructed interpretation resident in short term memory (STM); see Figure 9.

Descriptions of scenes, at various levels of detail, are stored in long term memory as a set of schema hierarchies [HAN78b]. A schema graph is a data structure defining an expected collection of objects, such as in a house scene, the expected visual attributes associated with the objects in the schema (each of which can have an associated schema), the expected relations among them, and control information for hypothesizing and verifying the presence of objects in the schema. This stored knowledge can be used to infer the presence and location of other objects, or verify uncertain hypotheses via spatial consistency of object labels. However, in order to use this knowledge there must be a basis for partial interpretations.

Global research issues that must be dealt with include the extraction and description of multimodal sensory data, the creation and maintenance of environmental models, the structure of cooperating control systems, and the development of knowledge structures necessary for integrating diverse sources of visual data into a comprehensive whole. Each of the following sections describes ongoing research aimed at the further development of the VISIONS system.

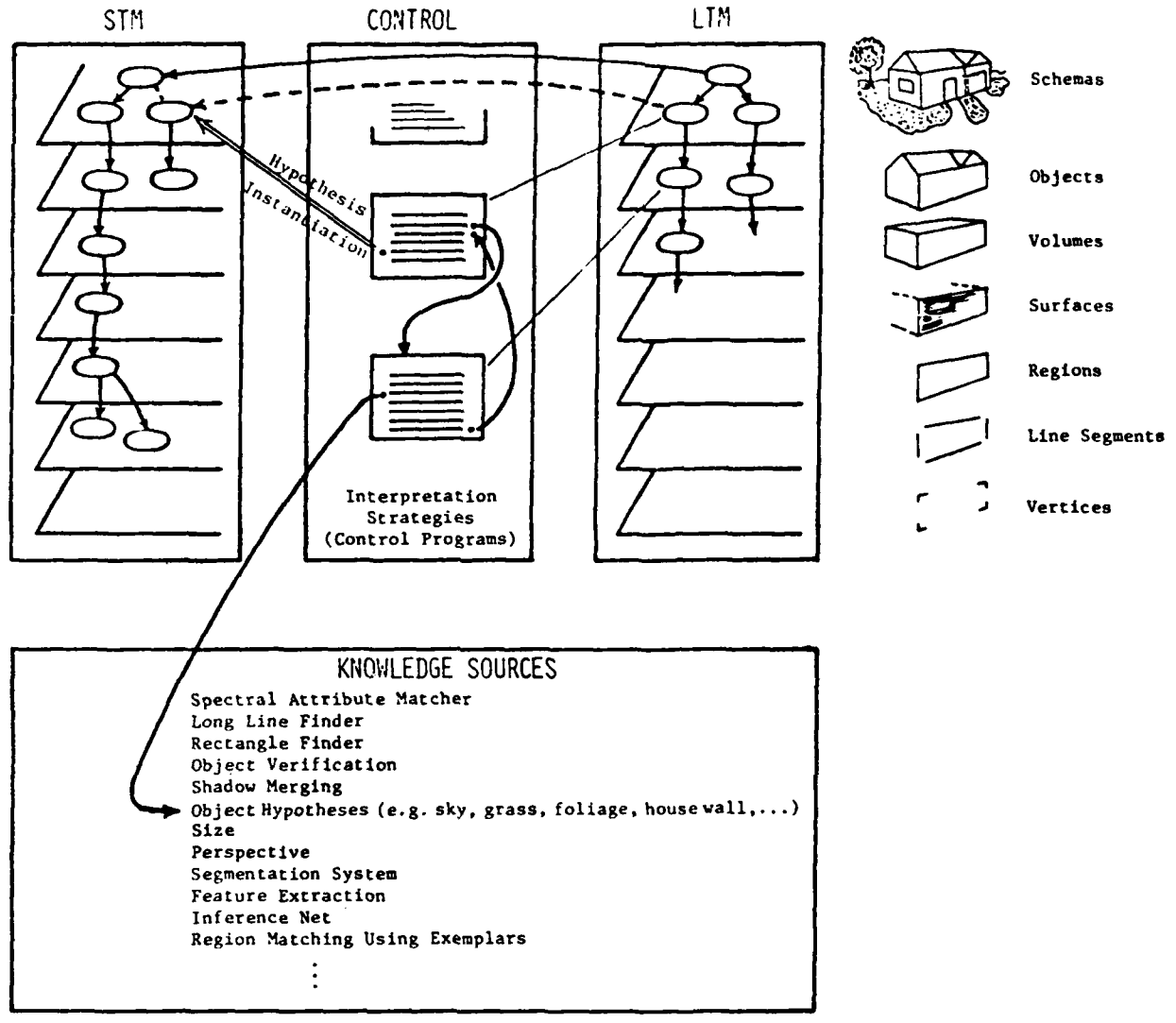


Figure 9. Block Diagram of Interpretation System.

Interpretation is the process of constructing a hierarchical semantic representation of the image contents in short term memory (STM). The semantic description is constructed under the control of interpretation strategies (control programs) associated with corresponding semantic entities in long term memory (LTM ≡ representation of world knowledge). The control programs hypothesize, verify, and instantiate nodes in STM on the basis of results returned by the knowledge sources and other interpretation strategies activated by the control program. Note that the actual structure of control and the interpretation process is more complicated than shown here; it is discussed in more detail in the text.

First efforts of the design of an integrated system were documented by Hanson and Riseman in [HAN78b], where only portions of the system were automatic. Parma, Hanson, and Riseman [PAR80] showed more complete results in the constrained case where knowledge was available about the particular house scene and the particular point of view (i.e., specific instance of a schema and a known viewpoint). The experiments demonstrated that processes could extract information of sufficient quality from an image to interpret complex natural scenes when the knowledge provided such very strong constraints. In addition, a methodology for generalizing the approach to less constrained situations was outlined. More recently the system has been demonstrated to operate on a set of complex house scenes of diverse appearance using only general knowledge of house scenes [WEY83].

III.2. Rule Based Object Hypotheses and Object Exemplar Strategies

Weymouth, Griffith, Hanson and Riseman [WEY83] have been developing a rule based image interpretation system which has been effective in labelling the regions of a set of complex outdoor scenes with their object identities. In the initial stages, there are few if any image hypotheses, and development of a partial interpretation must rely primarily on general knowledge of expected object characteristics that are independent of other hypotheses. The system utilizes a set of rules to select reliable object hypotheses as object "exemplars" in order to extend a partial interpretation.

The effectiveness of the interpretation process depends, in part, on an ability to extract image features which can be used to relate image events to semantic entities. Object hypothesis rules involve sets of partially redundant features each of which defines an area of feature space which represents a "vote" for an object. Thus, at the simplest level a rule is just a specification of a feature range which should be satisfied if an object is present. A set of simple rules can be combined (via any reasonable combining function) into a complex rule which is more reliable than any of the individual component rules; the premise is that many redundant rules allow any single rule to be unreliable. The features on which the rules can be based include color, texture, shape, size, image location, relationship to other objects, etc. For example, in an outdoor scene taken with a camera in standard position, one would expect grass to be of medium brightness, to have a significant green component, to embody a modest degree of texture, to be located *somewhere* in the lower portion of the image, etc. These expectations are translated into a rule which combines the results of many measurements into a confidence level that the region (or group of regions) represents grass (Figure 10a,b). By adding control information, an interpretation strategy can be constructed from the individual rules; an interpretation strategy describes how the various rules and processes can be used to hypothesize and/or verify semantic events (house scene, for example).

The extreme variations that occur across images can be compensated for somewhat by utilizing an adaptive strategy. This approach is based on the observation that the variation in the image appearance of objects (i.e., region feature measures across images) is much greater than object variations within an image. One such strategy extends a kernel interpretation derived through the selection of object exemplars, which are regions that represent the most

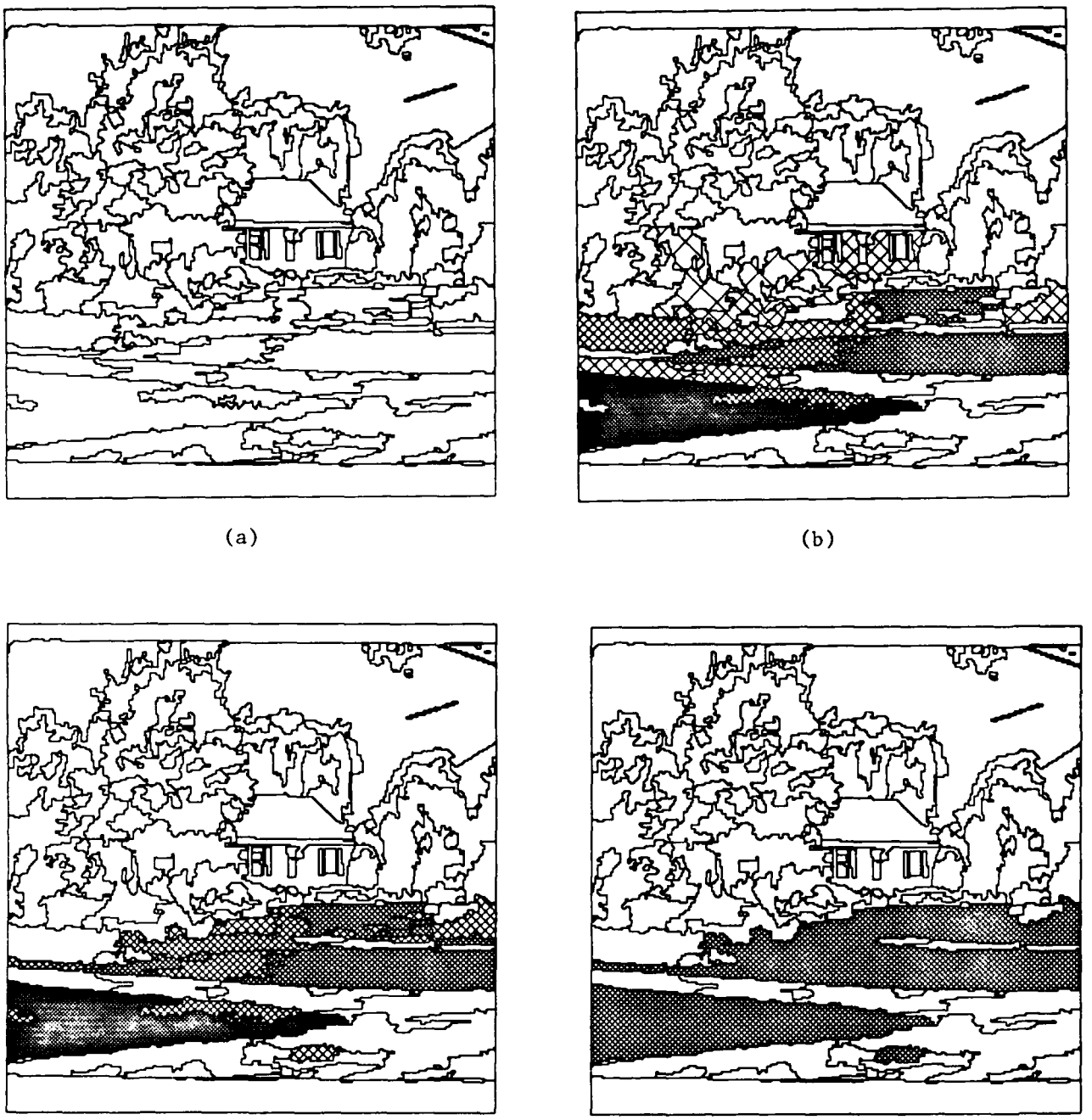


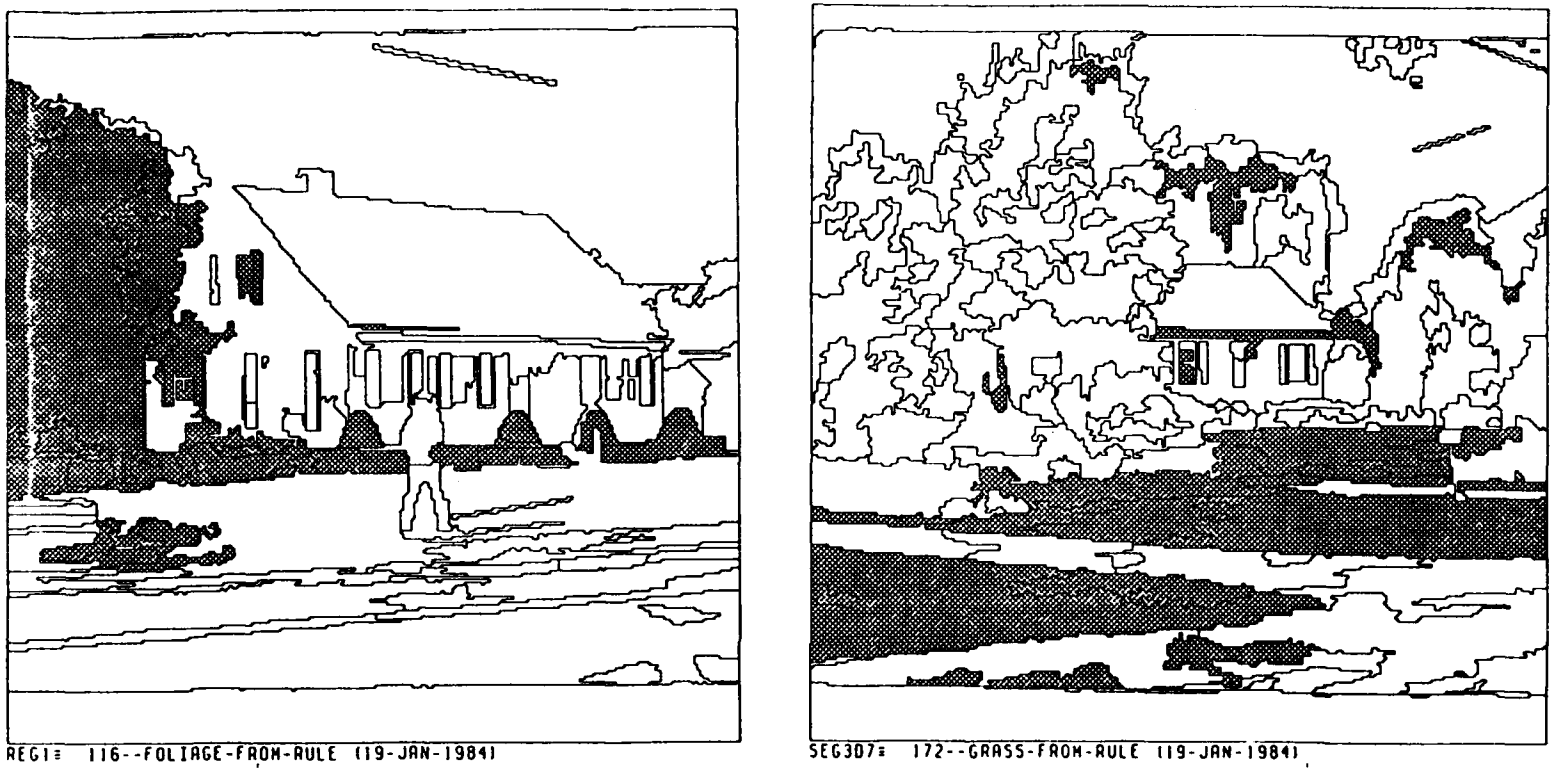
Figure 10. An Example Feature-Based Grass Rule.
(a) Original segmentation. (b) Regions hypothesized to be grass by means of a rule which matches region features to a description of grass stored in the knowledge network. The region with the best match is chosen as an image-specific exemplar and the features of this region are matched against the features of other regions. The density of the cross hatching is proportional to the match score. (c) Shows the grass regions obtained by selecting the best matches from (b). (d) The final hypotheses for grass regions obtained by merging the cross hatched regions in (c).

reliable image hypotheses of a general object class (Figure 10c,d). The use of exemplar strategies and other top-down strategies results in the extension of partial interpretations from "islands of reliability". Finally, a verification phase can be applied where relations between object hypotheses are examined for consistency. Thus, the interpretation is extended through matching and processing of region characteristics as well as semantic inference.

Experiments are being conducted on a set of fifteen "house scene" images. Thus far, we have been able to extract sky, grass, and foliage (often separating trees and bushes) from nine house images with reasonable effectiveness, and have been successful in identifying houses and their parts, including shutters (or windows), house wall and roof in three of these images (Figures 11 and 12).

III.3. Schemas and Schema-Directed Control

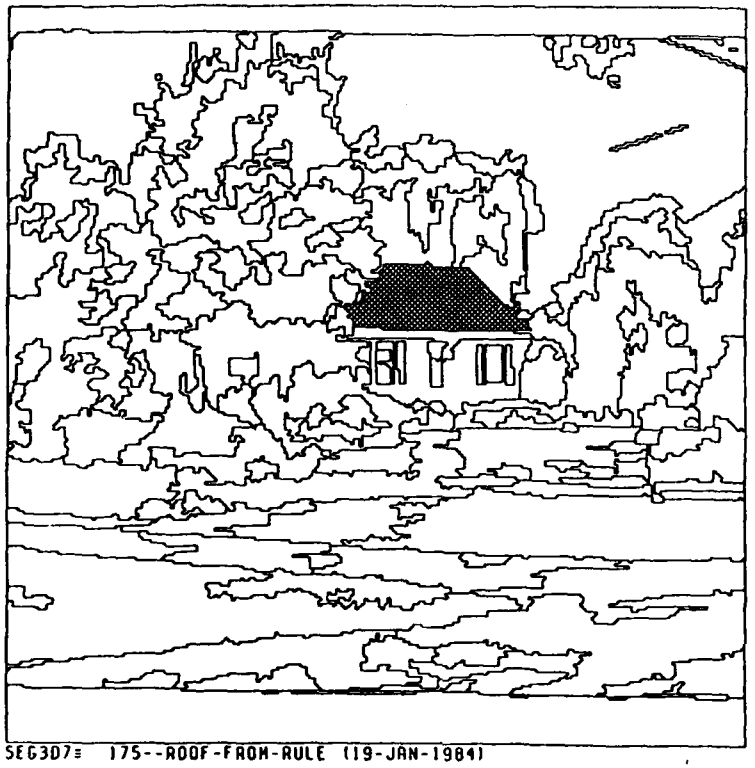
In order to effectively utilize the various forms of available knowledge during the interpretation, we have been examining an organization of the knowledge base based on schema hierarchies. These hierarchies combine descriptive and structural knowledge with the processes and strategies necessary to hypothesize and instantiate semantic scene entities to image events. Thus, a schema hierarchy is a data structure defining an expected collection of objects at various levels of semantic detail, such as a house scene, and appropriately detailed control strategies for applying processes that might detect and/or verify this set of objects in an image. For example, a house (in a house scene schema) has roof and house wall as sub-parts (each of which can have an associated schema), and the house wall has windows, shutters, and doors as sub-parts [WEY83]. Each schema node (e.g., house, house wall, and roof) has a structural description appropriate to the level of detail, including the expected visual attributes associated with the objects in the schema and the



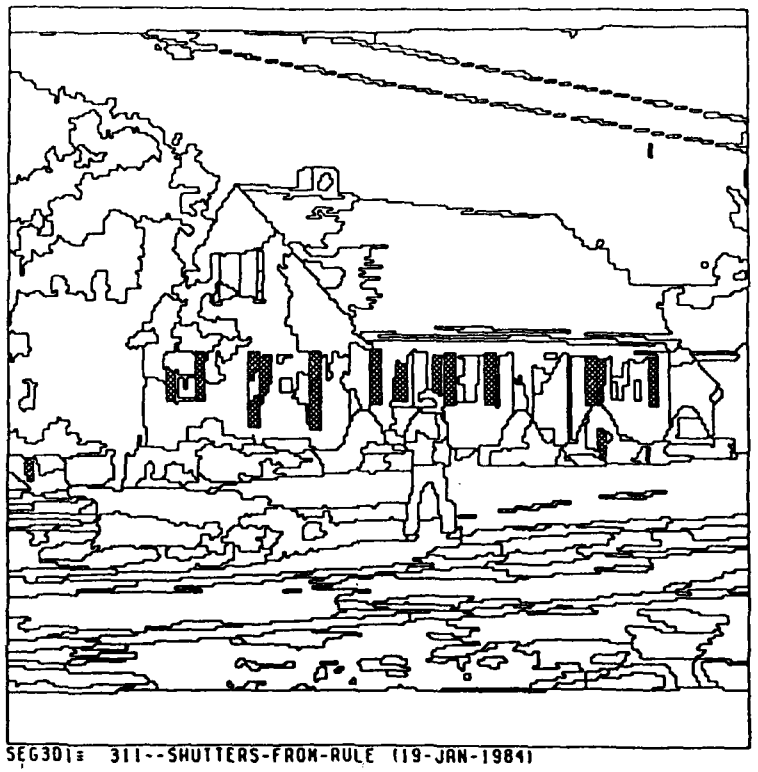
(a) (b)

Figure 11. Object Hypotheses from Interpretation Strategies Using Simple Rules.

These figures show isolated results from the interpretation strategies when applied to several images; the complete interpretations are shown in Figure 12. In each case, the segmentation used is one of those shown in Figures 4 and 5. (a) Foliage. (b) Grass (this result was obtained using a rule different from that used in Figure 10d).



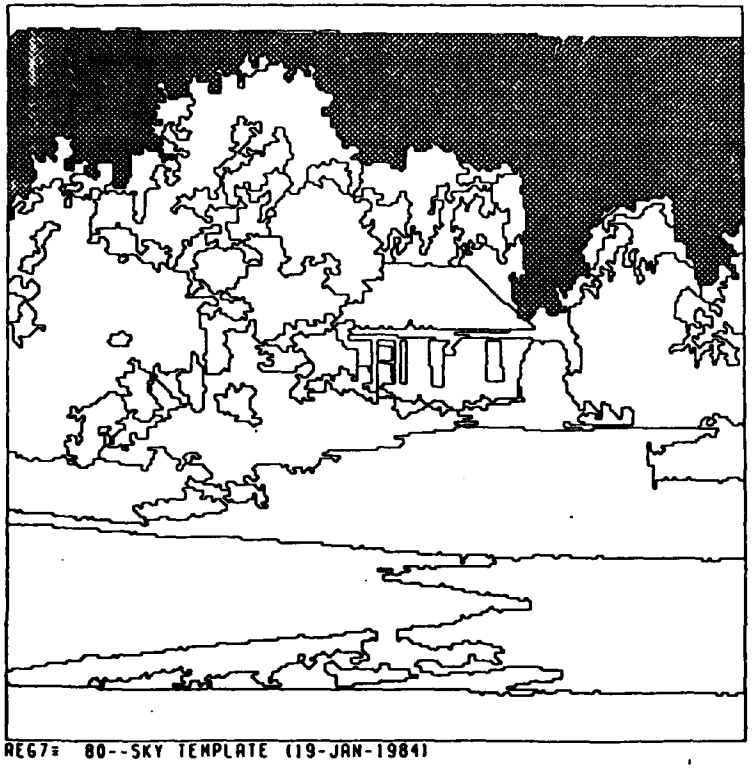
(c)



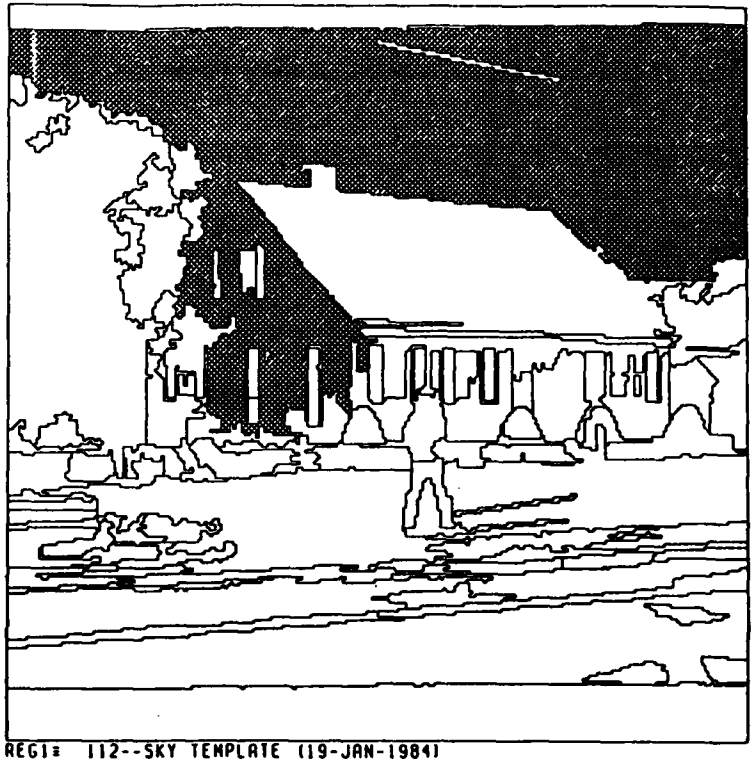
(d)

Figure 11, continued.

(c) House roof. (d) Shutters.



(e)



(f)

Figure 11, continued.

(e) Sky. (f) Sky. This last result combines both sky and house wall in the sky hypothesis; a similar result is obtained from a strategy for house wall. See Figure 12c,d,e.

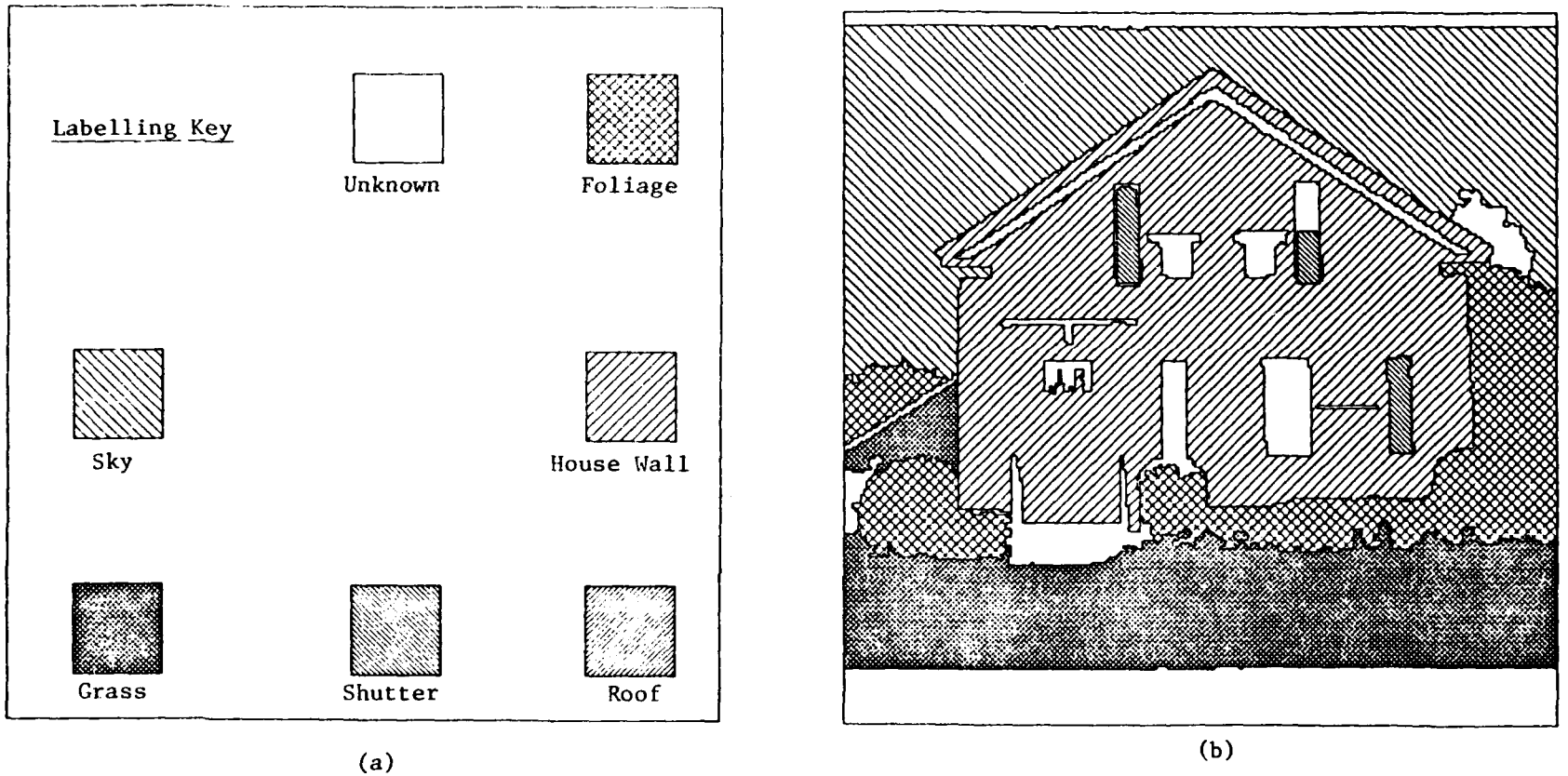
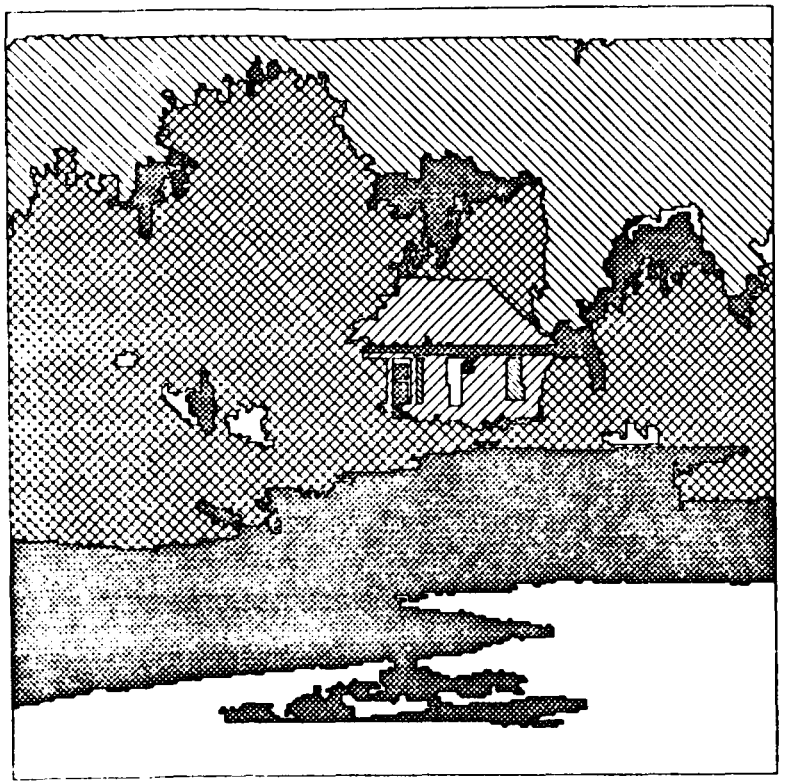
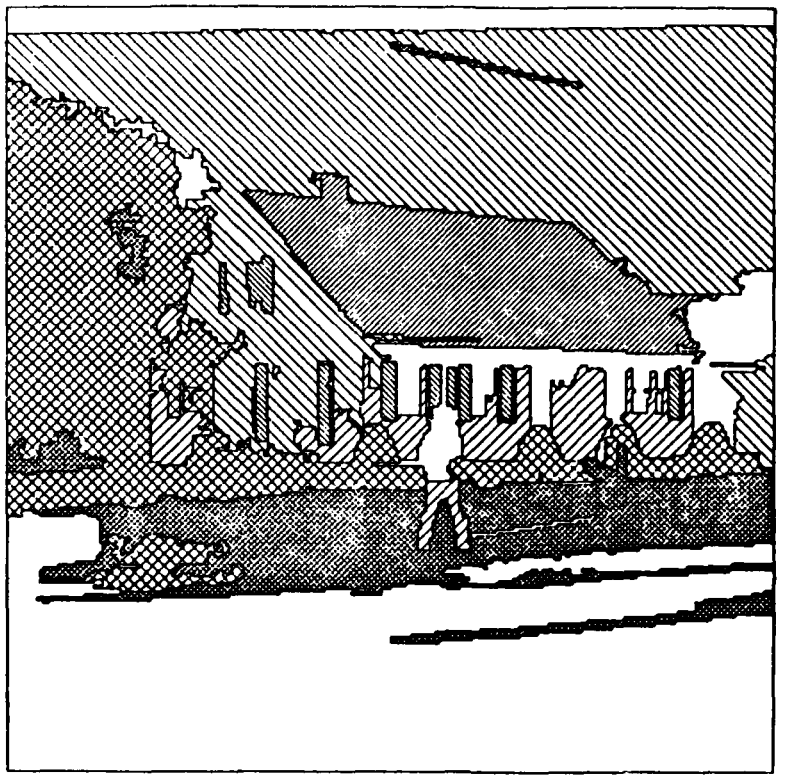


Figure 12. Final Interpretations.

These images show the final results obtained by combining the results of the interpretation strategies under the constraints generated from the knowledge base. (a) Interpretation key. (b) Interpretation results.



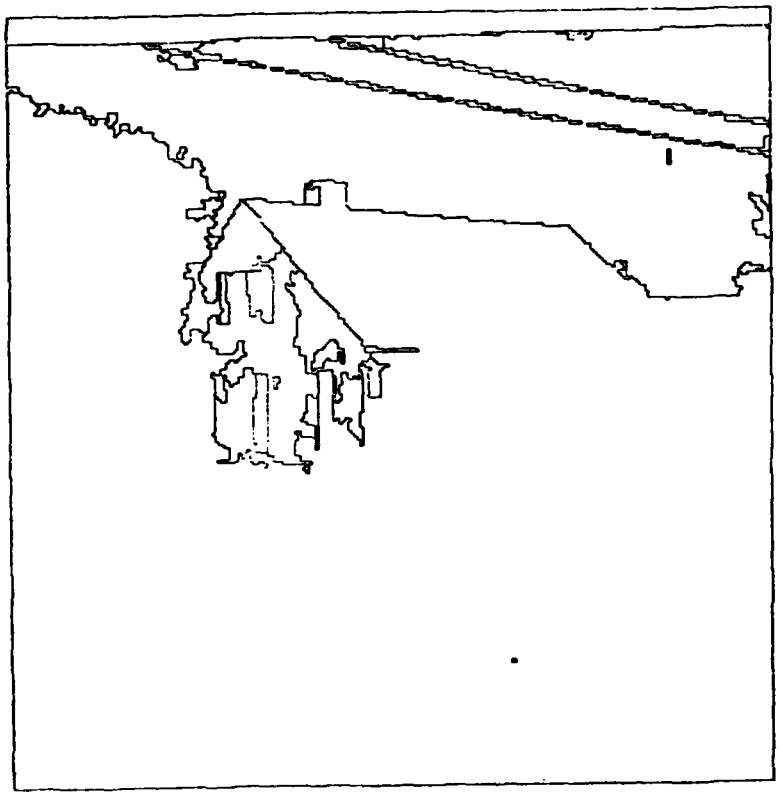
(c)



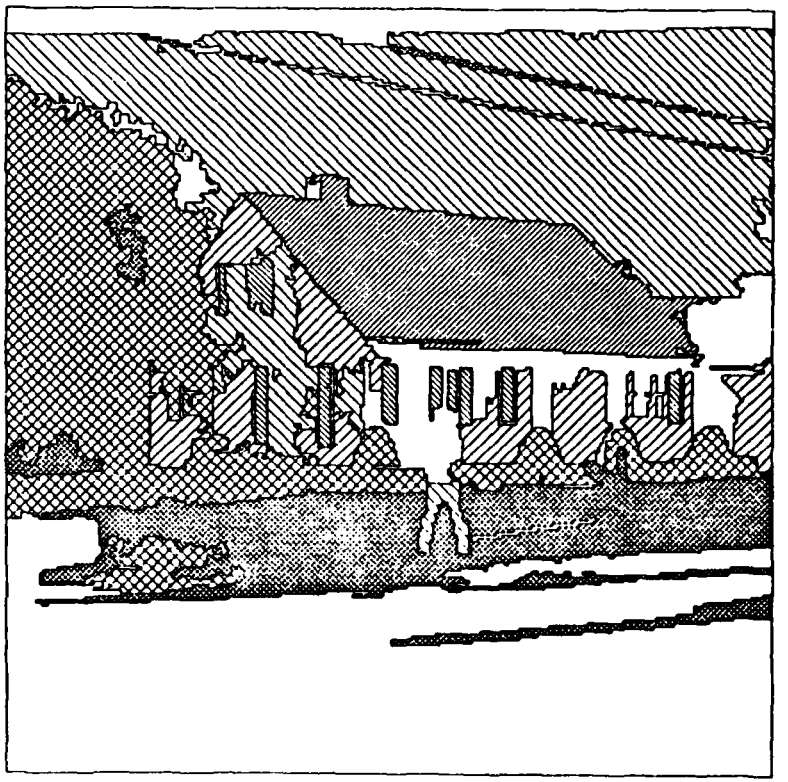
(d)

Figure 12, continued.

(c) Interpretation results. (d) In this image, the missing boundary between sky and wall results in a labelling conflict (the identification shown is sky; a second interpretation has this region labelled house wall).



(e)



(f)

Figure 12, continued.

(e) The sky/wall labelling conflict can be resolved in this image by resegmenting the region; the resulting segmentation shows internal structure since the processes are localized to the region. (f) Final interpretation after inserting the resegmented region and reinterpreting those areas related to the inserted region.

expected relations among them.

The schema control information involves a set of recognition and verification strategies called interpretation strategies (Figure 13). An interpretation strategy specifies how specific interpretation rules may be applied, and how combined results from multiple rules may be used to decide whether or not to "accept" (i.e., instantiate) an object hypothesis. The interpretation strategy thus represents both control local to the schema node and top-down control over the instantiation process. Note that the goal is not to expect these interpretation rules and strategies to always produce the correct hypotheses. Our philosophy is to allow incorrect, but reasonable, hypotheses to be made and to bring to bear other knowledge (such as various similarity measures and spatial constraints) to filter the incorrect hypotheses. Schemas divide the hierarchically organized long term memory into overlapping partitions at each level; each partition has a particular focus and with each there is an associated packet of information useful for top-down and bottom-up processing in the context of the schema node.

Our current focus is on developing and integrating interpretation strategies into the distributed control components of the schemas and demonstrating that under a reasonable set of constraints the control mechanism is adequate to correctly interpret the image. Since the appropriate schema for the scene is assumed known, we initially avoid some of the complicated control issues. This would show that if the correct schema can in fact be hypothesized, it could then be automatically verified and instantiated. A parallel effort is developing methods for hypothesizing, in a bottom-up fashion, a set of plausible schemas. The structure of the schema organization and effectiveness of the resulting interpretation system will be described in the forthcoming Ph.D. theses of Weymouth and Kohl, both expected in early 1985.

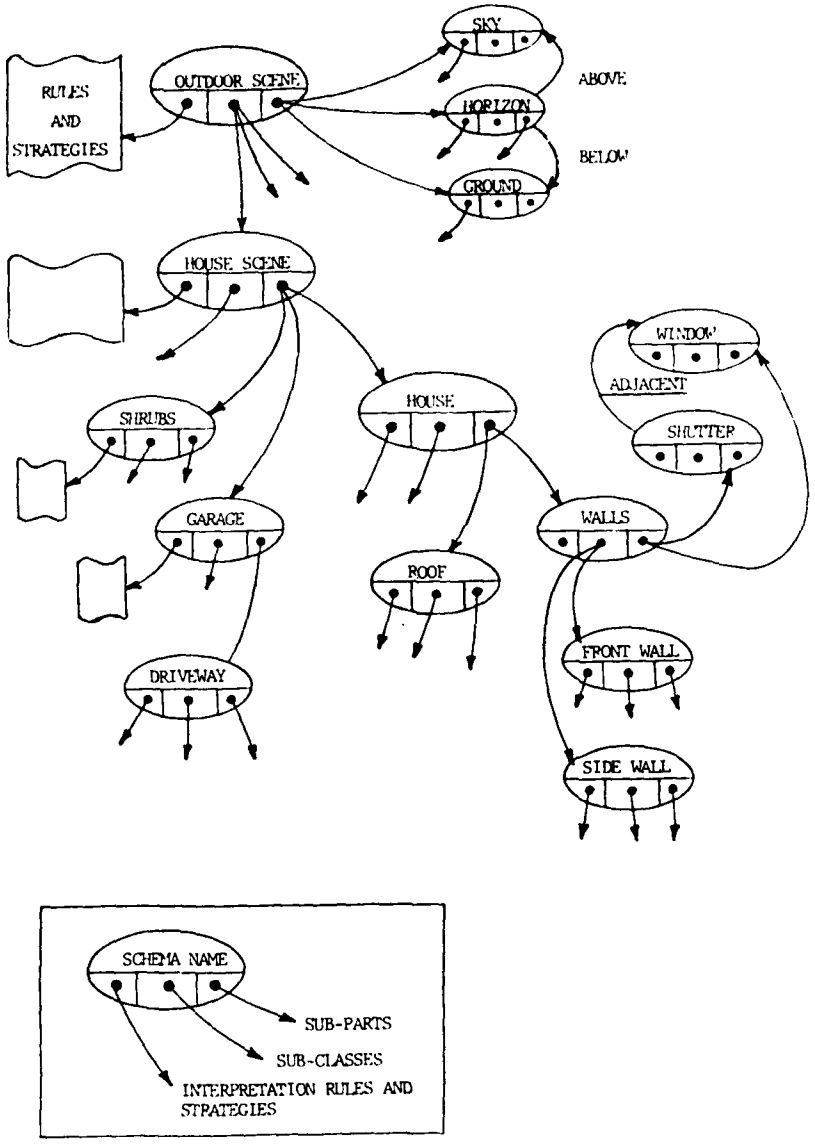


Figure 13. The Schema Representation of Long Term Memory.

Schema nodes encode the semantic identity of scenes, objects, and object parts as well as their structural and spatial decomposition into simpler components. Attached to each node are a set of interpretation rules and strategies which describe procedurally how particular image events may be hypothesized and verified to be instances of that semantic object.

III.4. Inferencing and the Inference Network

The construction of an image model is critically dependent on an ability to interpret the typically imperfect information provided by the various rules and interpretation strategies within the context of domain knowledge, system goals, and current hypotheses about the interpretation. An implicit assumption of our research is that the set of possible interpretations can be sufficiently constrained by some body of knowledge and inferences from the presence or absence of image features can be pooled correctly. A major factor contributing to this ambiguity is the degree to which the knowledge sources provide conflicting evidence. It has been shown [HAN78b, PAR80] that ambiguity arising from the lack of perfect information can be substantially reduced by obtaining partially redundant information from a variety of different sources. However, a major problem has been to develop mechanisms with some theoretical foundation that can take such unreliable and incomplete information and interpret it within the context of the available knowledge.

Some of the limitations of inferencing using Bayesian probability models are overcome using the Dempster-Shafer formalism for evidential reasoning, in which an explicit representation of partial ignorance is provided [SHA76]. The inferencing model allows "belief" or "confidence" in a proposition to be represented as a range within the [0,1] interval. The lower and upper bounds represent support and plausibility, respectively, of a proposition, while the width of the interval can be interpreted as ignorance.

Evidential information, extracted from the environment by modular sources of knowledge, enters these models in the form of probability "mass" distributions which are defined over sets of propositions common to both them and the model. These mass distributions are combined, relative to the

possibilities embodied in the model, through Dempster's rule of combination [DEM67]. The result is a new mass distribution representing the consensus of the information combined. This information is converted to the interval representation, and the model allows "inference" from those propositions it directly bears upon to those it indirectly bears upon (Figure 14). The apriori probabilities, frequently difficult or impossible to collect in artificial intelligence domains, but required by most other systems of inexact reasoning, are not needed. This form of evidential reasoning is more general than either a Boolean or Bayesian approach, yet it reduces to Boolean or Bayesian inferencing when the appropriate information is available.

Within the VISIONS system use of the DGMES model to reason about the environment involves four main steps:

1. Obtain evidence (for example, from the hypothesis rules) that tends to confirm or refute the truthfulness of hypotheses represented in a dependency graph (LTM).
2. Use Dempster's rule of combination [DEM67, DEM68] to pool the evidence into a form suitable for input to the inference engine.
3. Record the effect of the pooled evidence on hypotheses it bears directly upon.
4. Propagate the effect of the pooled evidence to the remaining hypotheses in the inference network by updating the confidence intervals associated with each semantic entity in LTM.
5. Determine the belief in each entity from the confidence interval.

There are many reasons why the evidential model is attractive. It separates the mechanisms for combining evidence from the mechanisms for making environmental inferences, allowing us to experiment with ways to combine data that are independent of the representation of domain knowledge. The model also does not require perfect information; however, if it is available, it can be easily integrated with existing information. The model can perform both data

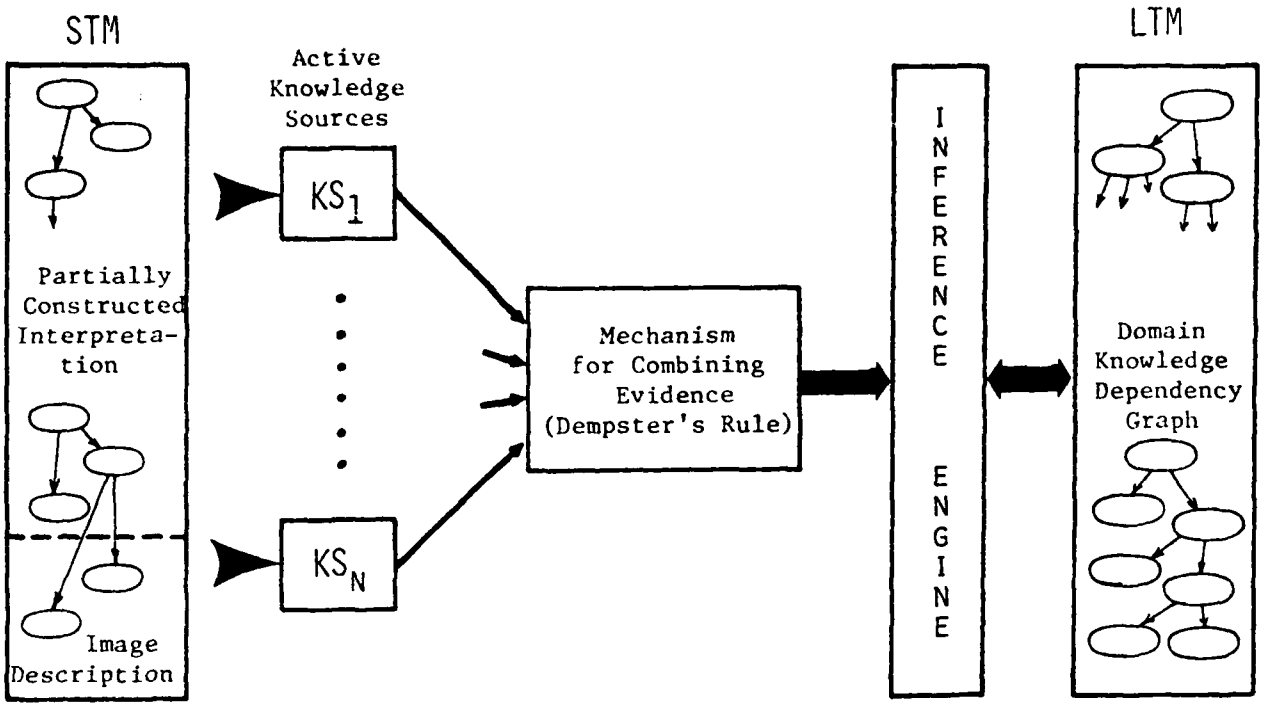


Figure 14. Architecture of the Inferencing Process.

Evidence for and against particular nodes (i.e., semantic concepts) in LTM is obtained via the application of modular knowledge sources to the partially completed interpretation in STM. Related evidence is combined and the results propagated through the domain knowledge represented as a hierarchically organized dependency graph in LTM. The resulting changes in the confidence levels attached to nodes in LTM may be used as the basis of a focus of attention mechanism and for system control.

and goal directed inferences over a single knowledge network. The theoretical foundation of the evidential model makes it easier to understand the relationship between the manipulation of environmental information and knowledge, and the performance of the system. Because of its formality it is easier to prove, if necessary, why the system performed the way it did, given some body of evidence and domain knowledge.

The inferencing system is implemented in GRASPER and has been applied to restricted cases of reasoning in the image interpretation domain [WES82]. We are currently examining ways in which the inference mechanism can be used to propagate the results of data-directed hypotheses through the long term knowledge structure leading towards schema instantiation; at the same time we are exploring the use of the same mechanisms to propagate downward through the knowledge representation toward activation of bottom-up processes. Both of these represent focus-of-attention mechanisms which can be used by the schema-driven interpretation strategies to determine how the current partial interpretation can be most profitably extended. Thus, we see the inference engine as a plausible connection between data-directed and goal-directed hypothesis formation and instantiation. Wesley [WES83] is extending this approach to distributed control over the interpretation process; the view is that by describing the available system resources, control over the interpretation process can be achieved using a set of very general, domain independent goals.

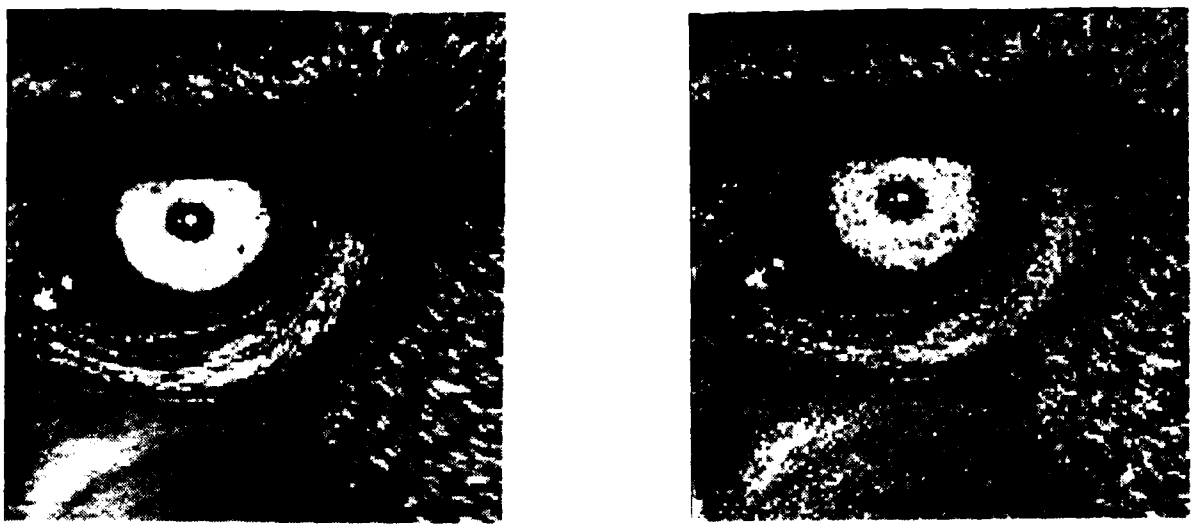
IV. Hierarchical Algorithms

IV.1. Feature Matching by Hierarchical Correlation

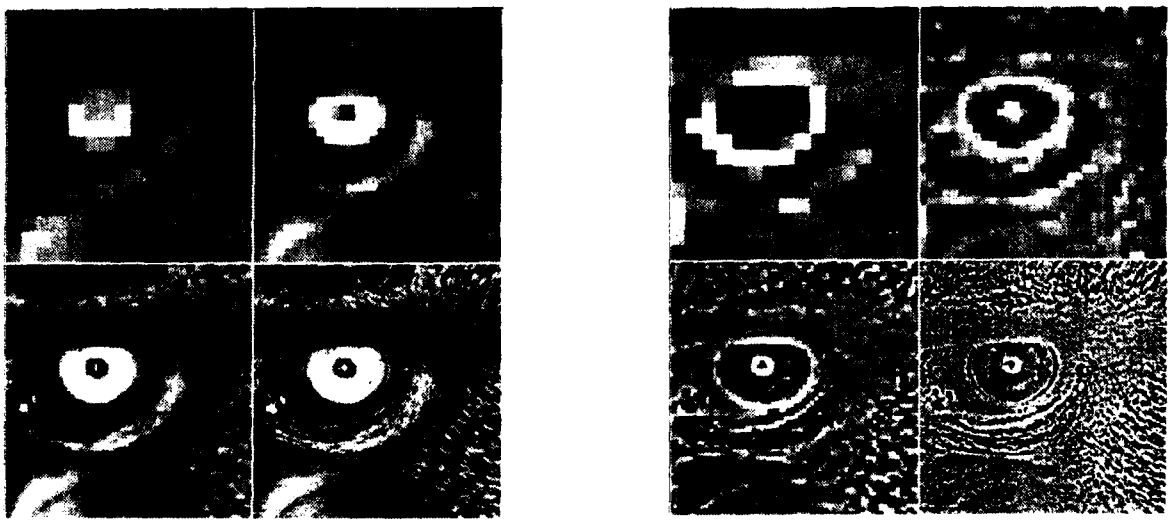
Feature matching algorithms are important in problems involving motion detection, image registration, and stereo vision. Hierarchical correlation provides a computationally efficient feature matching strategy. These algorithms can be implemented in hierarchical parallel hardware architectures, and they can also be implemented on a sequential machine to run very efficiently using a coarse to fine matching strategy.

Glazer, Reynolds, and Anandan [GLA83a] have developed a hierarchical matching algorithm that consists of matching band-passed versions of the images at different levels of resolution (Figure 15). The filters approximate convolution of a Laplacian and a Gaussian (del-squared-G) of different sizes. Alternative computational techniques for implementing the band-pass filter are being examined. One technique involves computing the del-squared-G at the finest level followed by a 4x4 Gaussian centered on 2x2 windows to reduce the resolution by a factor of two on each axis. These algorithms are computed in the processing cone [HAN80b] of the VISIONS Image Operating System [KOH82].

The matching is performed first on the low frequency structures occurring at the coarsest levels of the images, thus providing a coarse to fine strategy for matching higher frequency information at the levels below. This reduces the problem of false matches when, for example, there is high frequency texture with somewhat repetitive patterns. Thus, all useful information of the image is utilized at different levels: low frequency information at coarser levels and higher frequency information at finer levels.



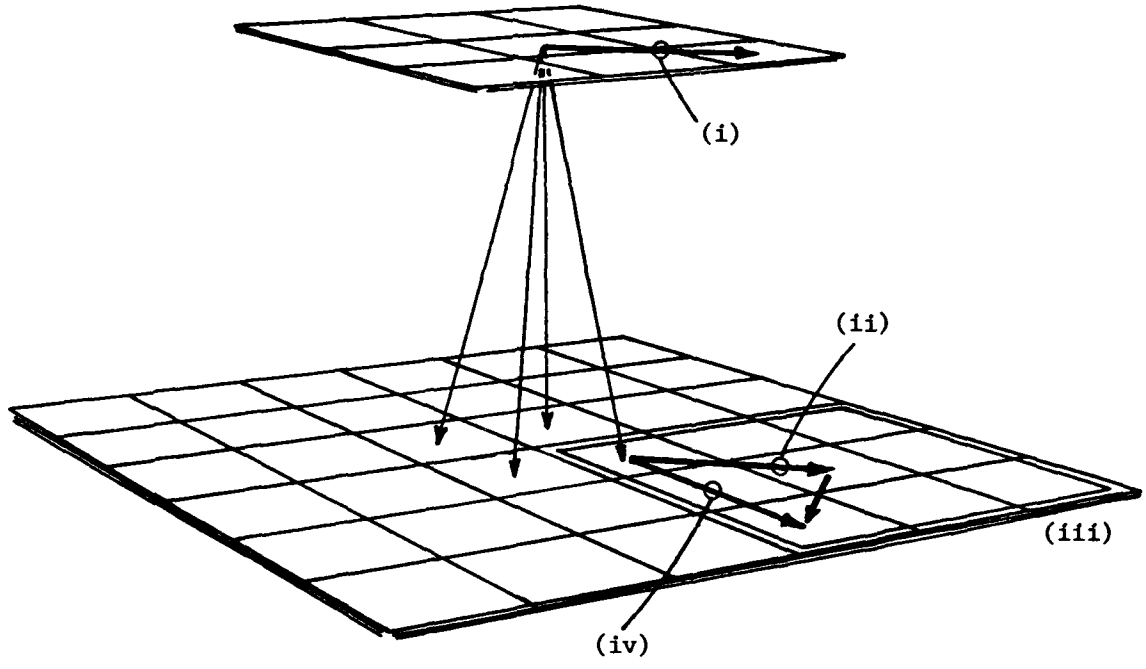
(a)



(b) (c)

Figure 15. Image Correlation by Hierarchical Feature Matching.

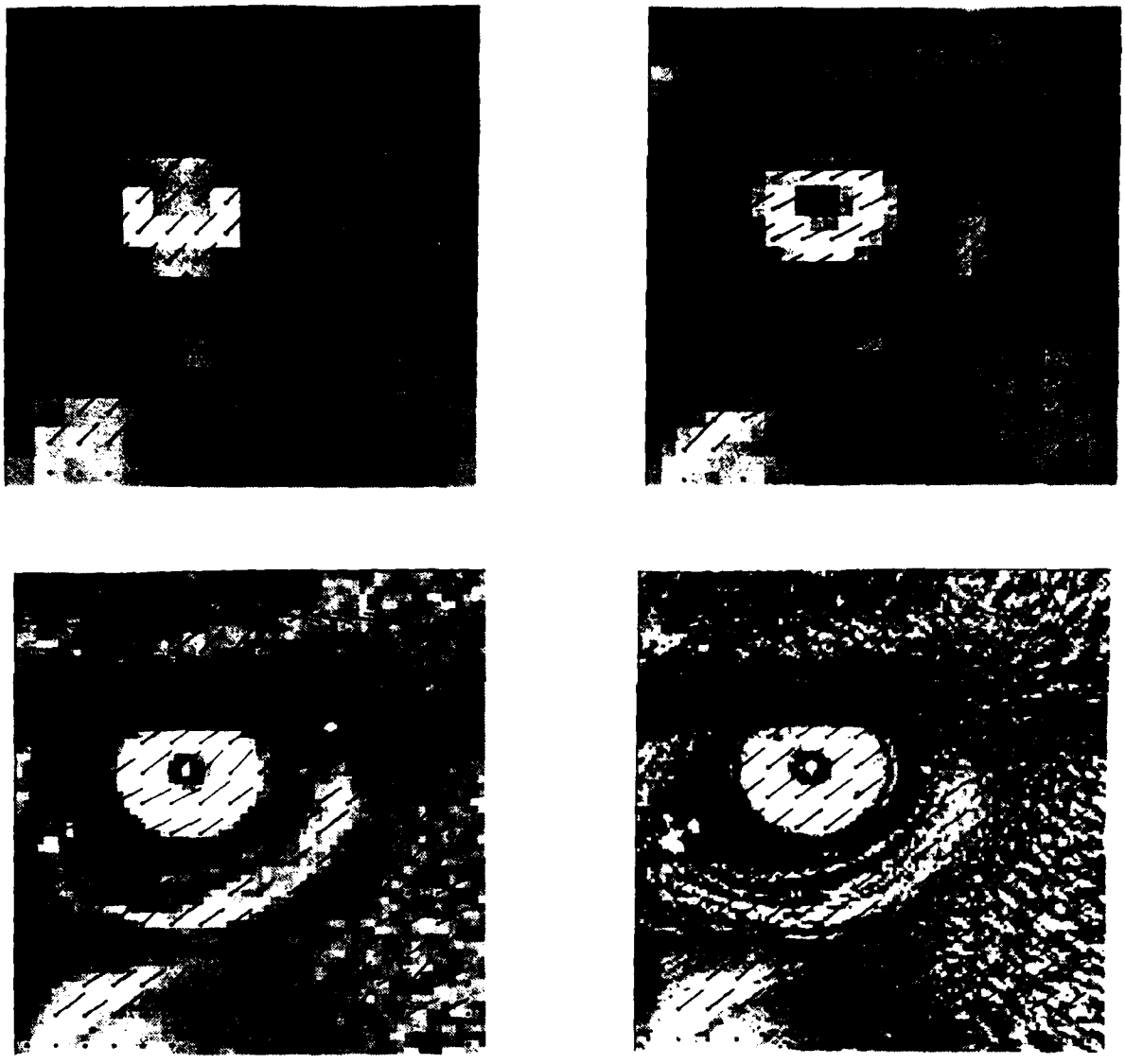
(a) Mandrill eye images used in the first experiment. The left side is a 128^2 piece of a larger mandrill image. The right side is the same image, translated 5 pixels up and 7 to the right, with white gaussian noise added (standard deviation = 10% of full range). (b) Low pass pyramid. Levels 4 through 7 of the low pass pyramid obtained from the mandrill eye image by applying the 4×4 reduction operator $[1\ 3\ 3\ 1] \times [1\ 3\ 3\ 1]^t$. [. . .] is a column vector, 'x' is the outer product operation, and 't' is the transpose operator. (c) Band pass pyramid. Levels 4 through 7 of the band pass pyramid obtained from the low pass pyramid in (b).



(d)

Figure 15, continued.

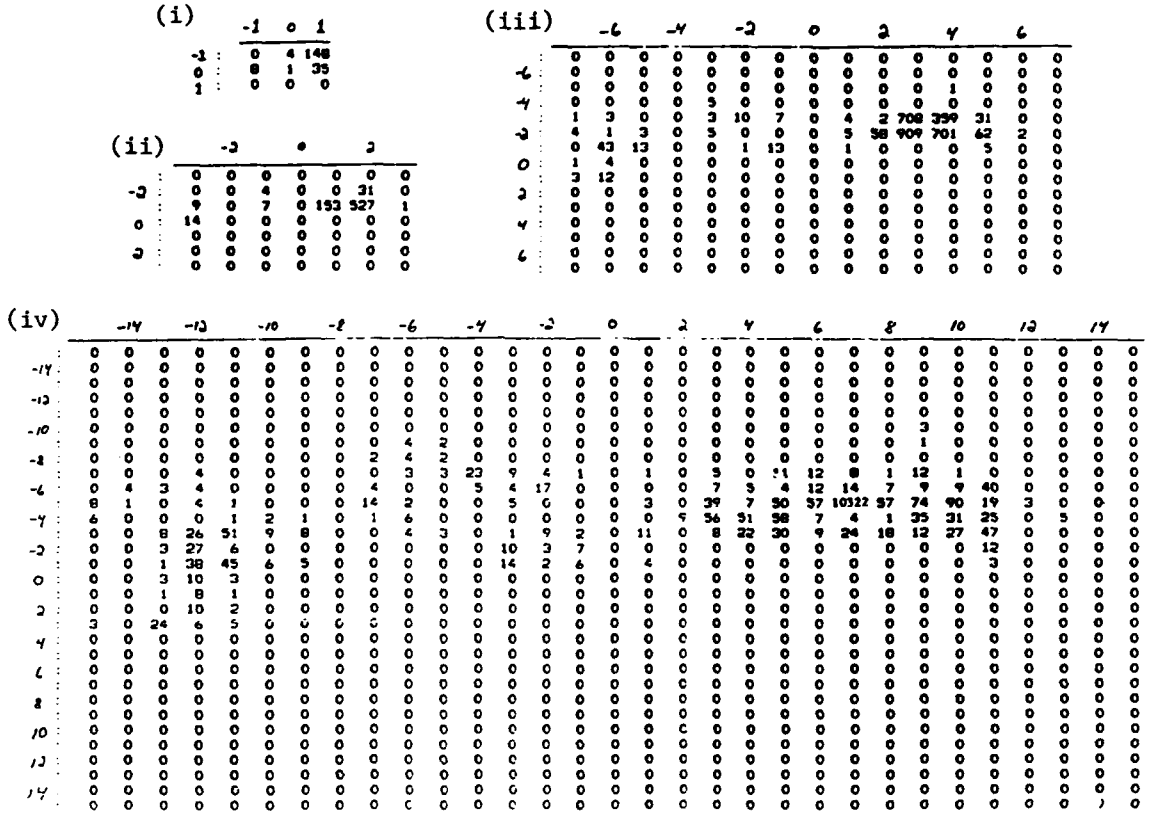
(d) 3 by 3 Search Algorithm.
- (i) Displacement vector at level N.
- (ii) Displacement vector projected to its four sons at level N+1 (only one of the four sons is shown).
- (iii) Search in a 3×3 area at level N+1 (search area shown in double lines).
- (iv) Updated displacement vector.



(e)

Figure 15, continued.

(e) Computed displacement vectors. The displacement vectors at levels 4 through 7 obtained in the Mandrill experiment. Only a 64^2 sample of vectors is shown at each level.



(e)

Figure 15, continued.

(e) Two-dimensional histograms of the row versus the column components of the displacements are shown for each of level 4 through 7 (i through iv). Note the high count found in the bucket corresponding to the correct displacement of (-5,7) in (iv). About 87% of the displacement values are exact, indicating that the hierarchical process is quite insensitive to noise.

The correlation strategy utilizes the observation that at some sufficiently coarse level, the maximum displacement of an image event between a pair of images is at most one pixel. This restricts the search at that level to a 3x3 area and provides an estimate of displacement within $\pm$ 1/2 pixel accuracy. The projection of this estimate to the next finer level provides an estimated displacement of $\pm$ 1 pixel and allows search to again be restricted to a 3x3 area, with the process repeating downward. There are two significant computational advantages of this process. The number of correlation matches considered is 9*logD instead of (2D+1)**2, where D is the maximum displacement possible at the finest level of resolution. In addition, an 8x8 correlation window size was used at all levels, and this would require a window of size (8D)**2 to capture the same amount of information in a single level of search across correlation positions.

The algorithm has shown in practical experiments to be reasonably effective in determining even small amounts of rotation, seems to be insensitive to noise, and of course is very efficient. The weakness of this algorithm is that errors made at coarser levels of resolution will be propagated downward. Anandan is currently investigating these issues using a measure of match confidence to control the use of coarser matches at finer levels of resolution. Experiments have shown that it may not be necessary to apply the algorithm to restricted sets of interesting points that have a high degree of distinctiveness (such as corners). Some experiments have shown correct results using all points, and thus the algorithm might work on an arbitrary sampling of points.

IV.2. Multilevel Relaxation Algorithms

Much work in low-level computer vision has involved the dense interpolation or approximation of sparsely-known or noisy data. A few examples are image smoothing, surface interpolation, and optic flow computation. A recent approach to these problems [GLA82] has formulated them in terms of optimization or constrained minimization. In general these techniques are equivalent to solving elliptic partial differential equations (PDE's) with boundary conditions and constraints.

In either formulation, these problems can be solved by a class of algorithms well suited to computer vision. It includes the Gauss-Seidel iterative method and assorted variants. These methods are local, parallel, and distributed, attributes which make them ideal for implementation on locally connected parallel processors. They have one attribute, however, that currently limits their applicability. The number of iterations required for convergence is often very high -- on the order of O(d**n), where 'd' is the distance (in nodes) that information has to travel and 'n' is the order of the PDE's being solved. Their slowness is due to the fact that solutions which must satisfy a global condition (the variational problem) are arrived at by the local propagation of information.

In the problem domain of elliptic PDE's, slowness has been overcome by using multi-level relaxation algorithms. Multi-level relaxation is an algorithmic extension of iterative relaxation designed to overcome asympototically slow convergence. By representing the spatial domain at multiple levels of resolution (in registration) these algorithms apply the basic local iterative update to a range of neighborhood sizes. Local updates on coarser grids introduce a more global propagation of information. At each level

the problem is solved in a different spatial bandwidth. Thus, the various processing levels cooperate to compute the final result, which is represented at the highest resolution level. The number of iterations required is of order O(d). In experiments involving the solution to Laplace's equation, with fixed boundary conditions, a multi-resolution algorithm was shown to be over an order of magnitude faster.

We have recently applied a multilevel relaxation algorithm to the problem of computing optic flow from dynamic images. A variational problem was established, Euler's equations were derived, and a Gauss-Seidel iterative relaxation algorithm was formulated. This algorithm was then extended to a multilevel relaxation algorithm. Experiments indicate significantly faster convergence (4 to 10 times for the problem chosen) for the multi-resolution algorithm [GLA82].

IV.3. Hierarchical Segmentation and Focus of Attention Mechanisms

The detailed examination, segmentation, and understanding of high resolution digital images represents a severe computational load for current computers. One technique for reducing the overall computational requirements involves selectively focussing on relevant portions of an image and ignoring irrelevant portions. The specification of relevancy implies some external model which represents a description of those areas or objects that are of potential interest and to which computational resources may most fruitfully be applied. The most suitable method for applying such selective processing to high resolution imagery is the multi-resolution, or pyramid, technique (Section I.2). From the original, large-scale, full resolution image is constructed a progression of smaller and smaller images, each covering the same extent, but at

successively coarser resolution.

In this section we describe some recent experiments using a hierarchical segmentation algorithm and focus of attention mechanism for locating buildings, roads, and airports in a high-resolution monochromatic aerial image. The approach involves formulating the segmentation and feature extraction algorithms described in Section II as hierarchical algorithms within the processing cone. The focus of this section is on the segmentation processes; more complete interpretation results may be found in [REY84a,b].

The general idea is to use the Nagin-Kohler region segmentation algorithm and the Burns linear feature extraction algorithm (see Section II.3) as the primary low-level processes used to drive the bottom-up component of a hierarchical localized segmentation process. The feature extraction process yields a low-level representation of the data and an evidential-based inference net (see Section III.4) is used to transform this data to an intermediate level of representation within long-term memory. This intermediate level of representation in turn allows the multiresolution segmentation algorithms to be focussed and selectively applied to areas of interest in the image. We are investigating the effectiveness of directing the system to look only in areas where a coarse level segmentation yields a hypothesis that an object of the sort we are looking for exists.

The hierarchical segmentation process can be summarized as follows. First the local histogram segmentation and the linear feature extraction algorithm are applied at a coarse level of resolution. Properties of the regions and lines are computed and stored in short-term memory. An inferencing network representing long-term memory is then invoked and each region then yields a support and plausibility [WES82] that it is a candidate region for one of the

objects we are looking for. The Nagin-Kohler algorithm and the Burns algorithm are then applied at a finer level of resolution, but only within the sectors which intersect the projections to the finer level of the candidate regions which have high support. At this finer level of resolution the representation of the object is of a different form and may involve more expensive combinations of the region and line attributes. However the inferencing process will now only be applied to a small subset of the image.

For high-resolution imagery, the computational advantage of this approach is significant. For example if we assume that even 2/3 of the possible sectors are used at each level, then only 1/5 of the image is being examined 4 levels down. In the case that only 1/4 of the sectors are selected, only 1/256th of the image is being searched 4 levels down. In addition, the knowledge base can be structured in terms of hierarchical resolution models. Long-term memory can be structured to be level dependent, so that although at finer levels of resolution the description of objects and hence the inferencing process is more complicated, the system is looking at only a small fraction of the image. Thus the computational complexity of the process can be kept within reasonable bounds.

While this work is in an early stage of development, Figure 16 shows some preliminary results on using a combination of the region segmentation and linear feature algorithms to locate areas containing runways. Figure 16(a) is a 512x512 subimage of the original 4096x4096 image. Figure 16(b) shows the same image reduced to level 7 (128x128) in the processing cone using a simple operator which computes the average of 2x2 blocks of pixels. Figure 16(c) shows the linear features extracted, 16(d) shows the lines resulting from the application of a simple filtering operation on length and contrast. These lines are then clustered in Hough space on the basis of orientation and then projected

onto the region segmentation at the next higher level of resolution (in this case the one obtained at level 8). Figure 16(e) shows the mask resulting from the intersection of the linear features with regions which bordered the selected lines and which satisfied a liberal size constraint. Figure 16(f) shows the segmentation mask obtained by selecting all of the sectors defined by the Nagin-Kohler algorithm which intersect the regions obtained in the previous step. This mask is used in turn to activate the region segmentation algorithm at the next finer resolution level in the processing cone and the whole process is repeated. Figure 16(g) shows the resegmentation at level 9 under the mask.

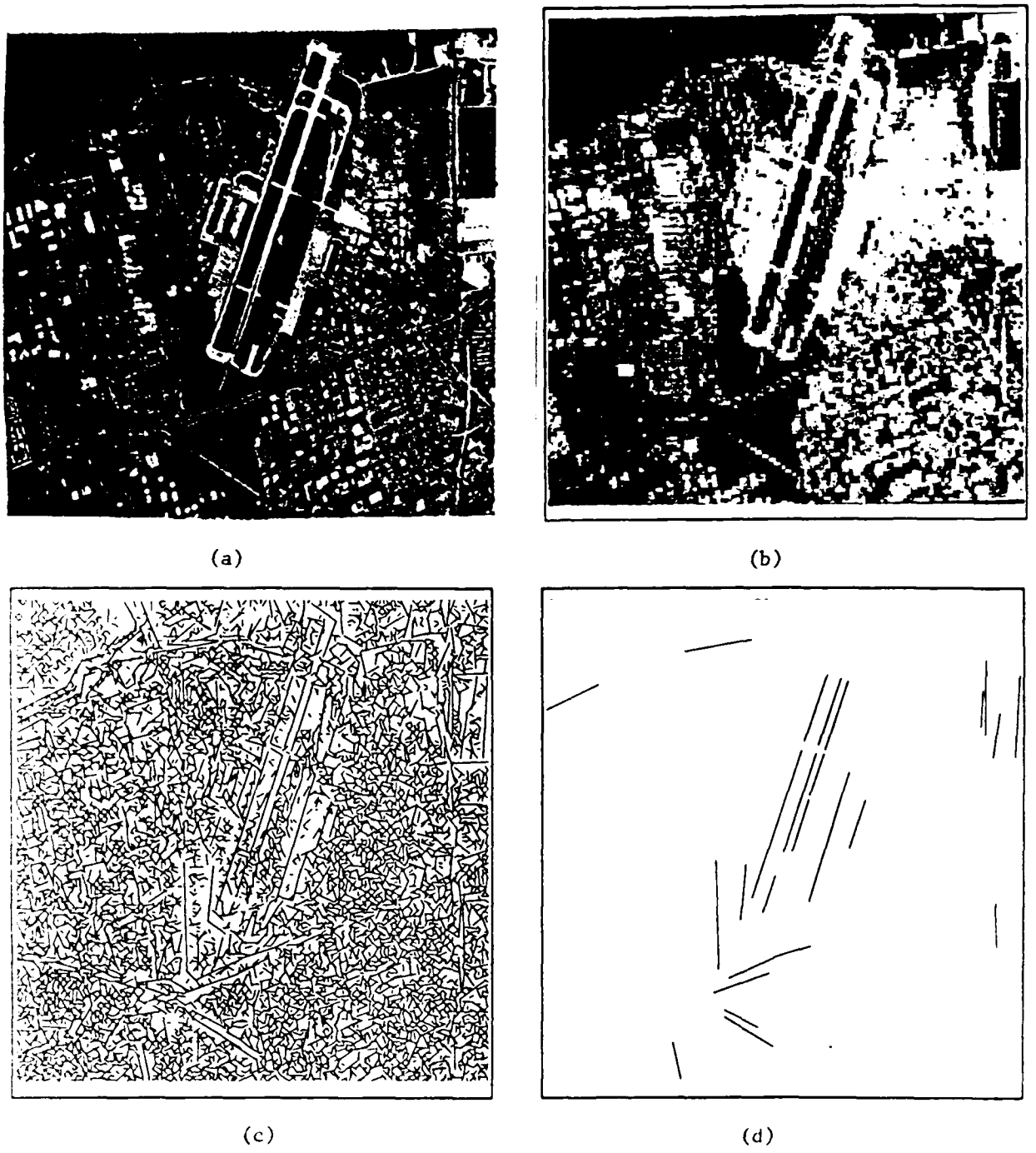
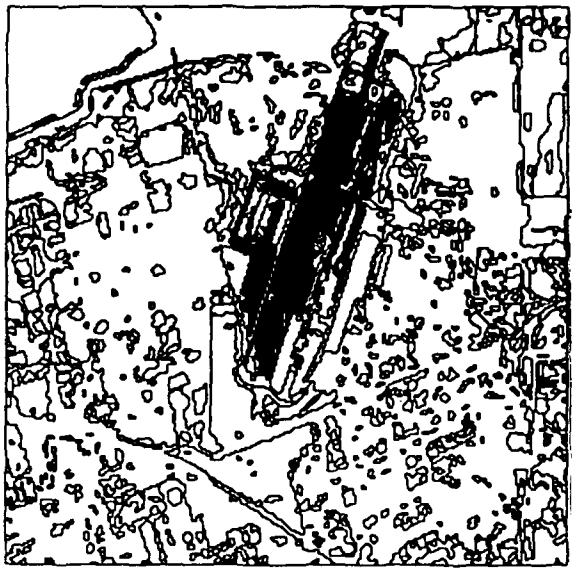
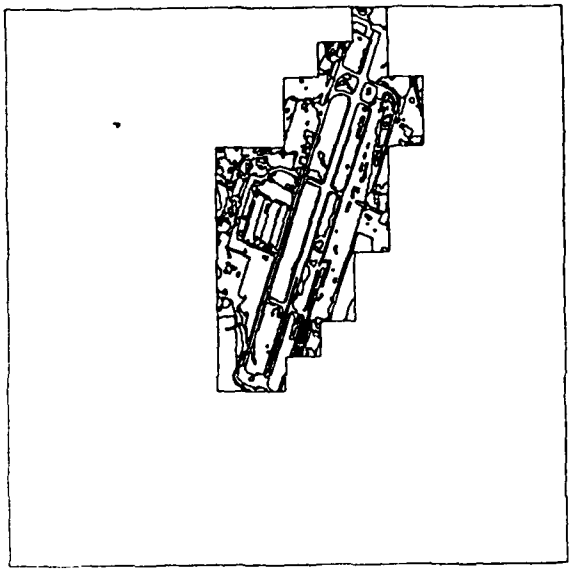


Figure 16. Hierarchical Segmentation Using a Combined Region/Line Representation. (a) 512×512 portion of a 4096×4096 image. (b) The image reduced to level 7 (128×128) in the processing cone by an averaging operator. (c) Results from the line algorithm described in Section II.2 applied to the reduced resolution image. (d) Lines remaining after filtering the lines on the basis of length (>9 pixels) and contrast (>12 gray levels).



(e)

(f)



(g)

Figure 16, continued.

(e) After clustering the lines in Hough (ρ,θ) space, the edge support regions which intersected intensity segmentation regions (shown as a crosshatched region) are used to form a segmentation mask at the next higher resolution level in the cone.
(f) The mask formed by selecting all the Nagin-Kohler sectors which intersect the regions.
(g) Segmentation results obtained by applying the Nagin-Kohler algorithm to that portion of the image intersecting the mask.

V. Motion Processing for Recovery of Environmental Depth

V.1. Introduction

The primary goal of the work in motion processing is the recovery of the motion parameters of the sensor and each independently moving object. The computation of environmental depth of visible surfaces follows in a rather straightforward manner. This has generally involved two stages of processing: computation of a feature displacement field, followed by inference of motion parameters and environmental depth [PRA79, PRA83, WIL80, WIL81]. We present several algorithms for performing this computation in independent stages, and for several restricted cases of sensor motion, some new alternatives for combining the two stages in a robust manner.

The set of image displacements from two or more images is an approximation to optic flow. During this stage of the processing one faces the well-known correspondence problem, which involves matching of corresponding image points of an environmental feature in the pair of images. The second stage involves inference of environmental information from the optic flow or the displacement field. This reduces to the problem of separating the translational and rotational components of the flow field.

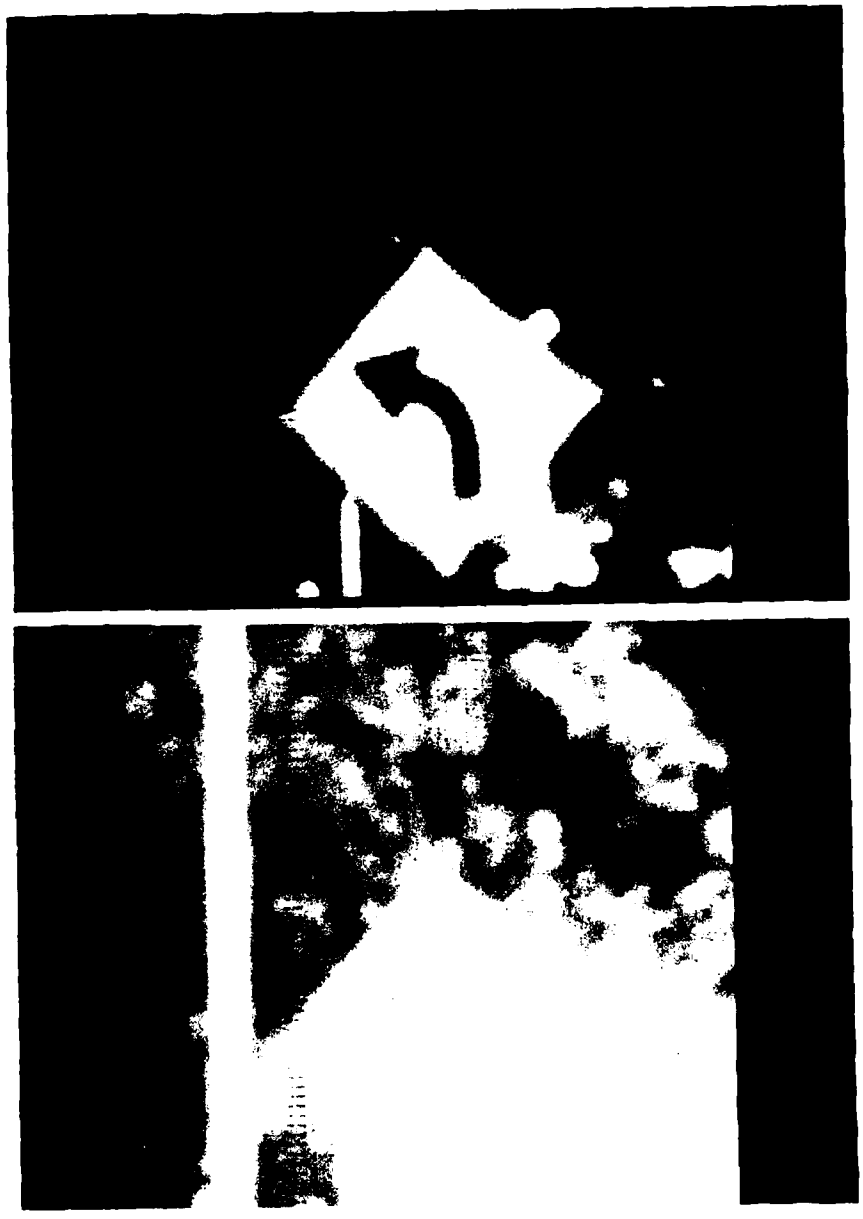
Rotation of the sensor induces image displacements that are a function only of the rotational parameters and image position; in particular the feature displacement between images is not a function of the depth of its environmental surface point.

The translational motion of the sensor carries all of the environmental cues. For purely translational motion, the image displacement paths are determined by radial flow lines emanating from a single point in the image plane, that is the intersection of the translational axis with the image plane (also referred to as the focus of expansion - FOE). The size of displacements along these paths are a function of environmental depth and distance from the FOE. Thus, the problem of general motion becomes one of decomposing the rotational and translational effects of motion, and then using the image displacements from the instantaneous component of translational motion to compute depth.

V.2. Restricted Cases of Sensor Motion

Our primary technique for depth inference is based on Lawton's doctoral dissertation [LAW84]. He has shown that in the cases of restricted sensor motion - pure translation, pure rotation, and motion constrained to a plane - the correspondence problem can be bypassed, or at least simplified, by combining the computation of the motion parameters with the determination of image displacements.

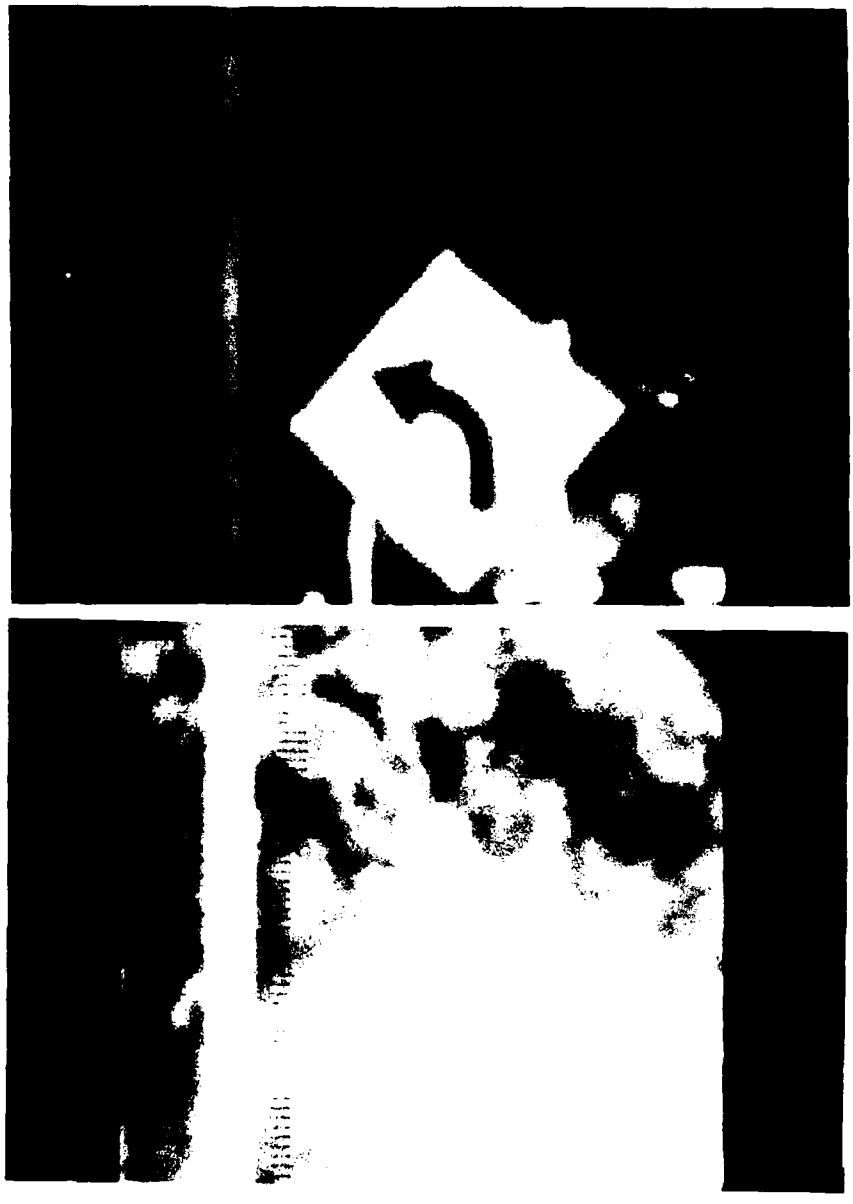
Let us illustrate with the case of pure translational motion [LAW83c, LAW84]; see Figure 17. There are two unknown sensor parameters which can be specified by the intersection of the translation axis with the image plane (the FOE). For a given FOE, the flow lines emanate radially from this point, and therefore the matching of an image point in one frame to its new position in the second frame has been reduced to a one-dimensional search along the straight line between the FOE and the image point. While there may still be spurious high correlations possible, the number of incorrect good matches will be greatly



(a)

Figure 17. Recovery of Translational Motion Parameters.

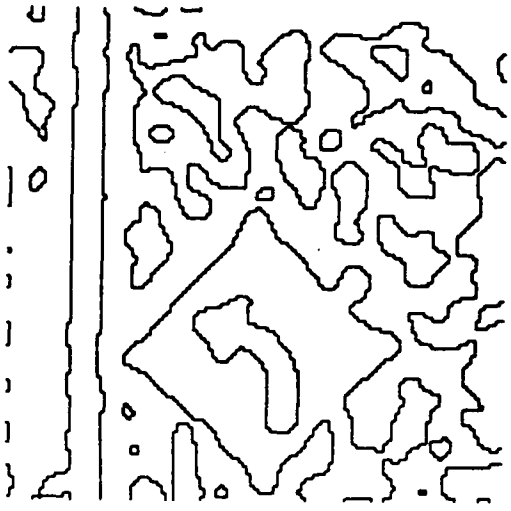
(a) First image from a sequence from a camera translating down a road. The upper image has the intensity values normalized across the entire image. The lower image uses a restricted range of intensity values to show the dark, low contrast tree texture.



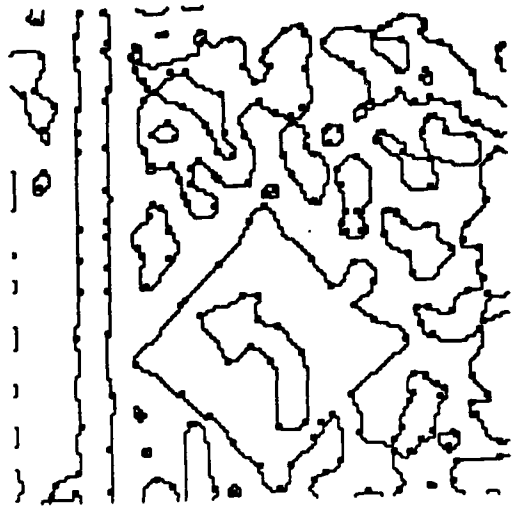
(b)

Figure 17, continued.

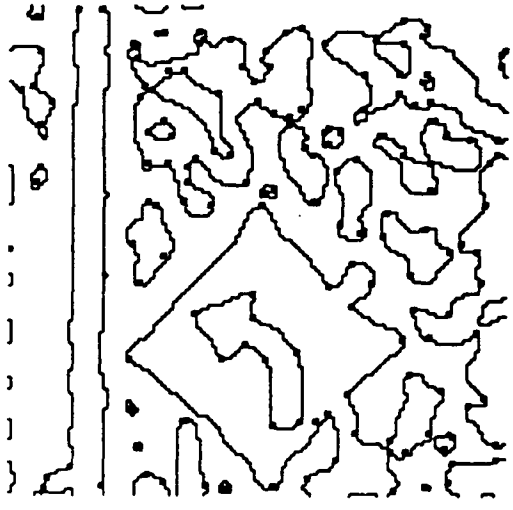
(b) Second image. The upper image has the intensity values normalized across the entire image. The lower image uses a restricted range of intensity values to show the dark, low contrast tree texture.



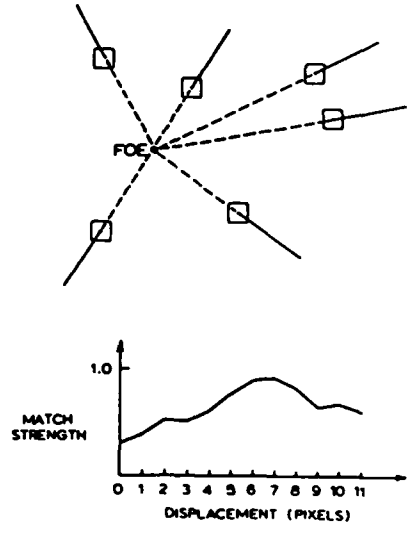
(c)



(d)



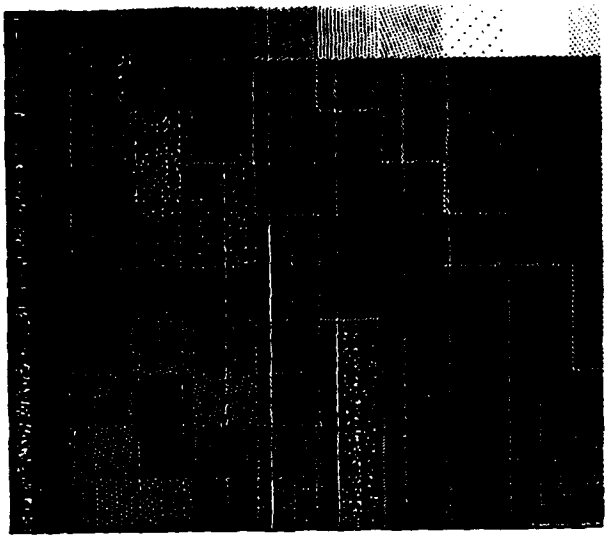
(e)



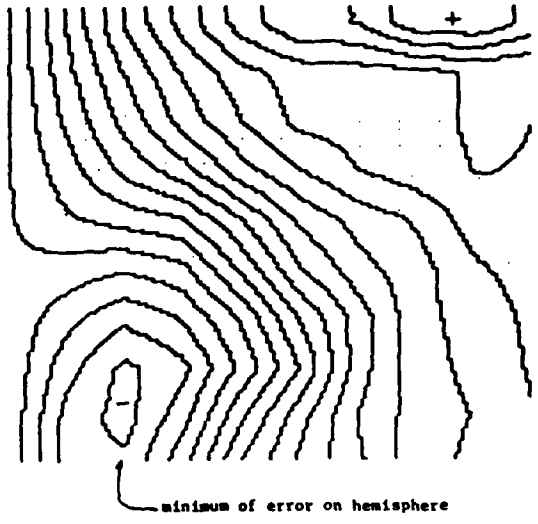
(f)

Figure 17, continued.

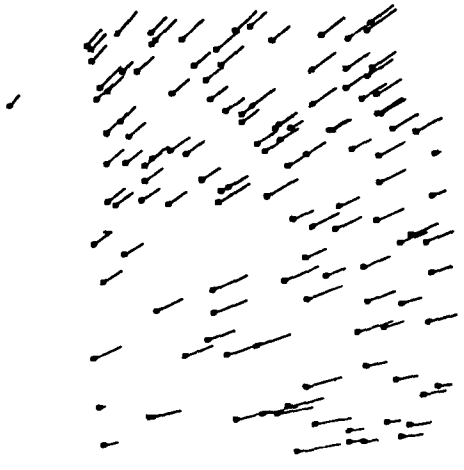
(c) Zero crossings extracted from image in 18a. (d) Interesting points extracted from figure 18a along the contour in figure 18c using a local interest measure. (e) Interesting points obtained from 18d by thresholding on curvature. (f) Using the extracted points, a search process minimizes an error measure which reflects the extent of feature mismatch with the next image in the sequence along displacement paths determined by a hypothesized translational axis.



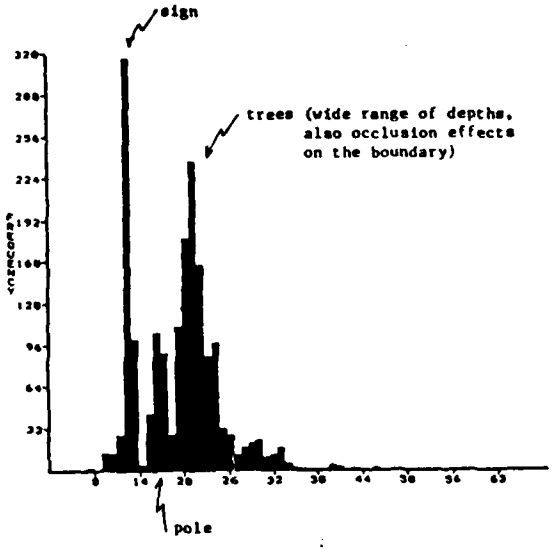
(g)



(h)



(i)

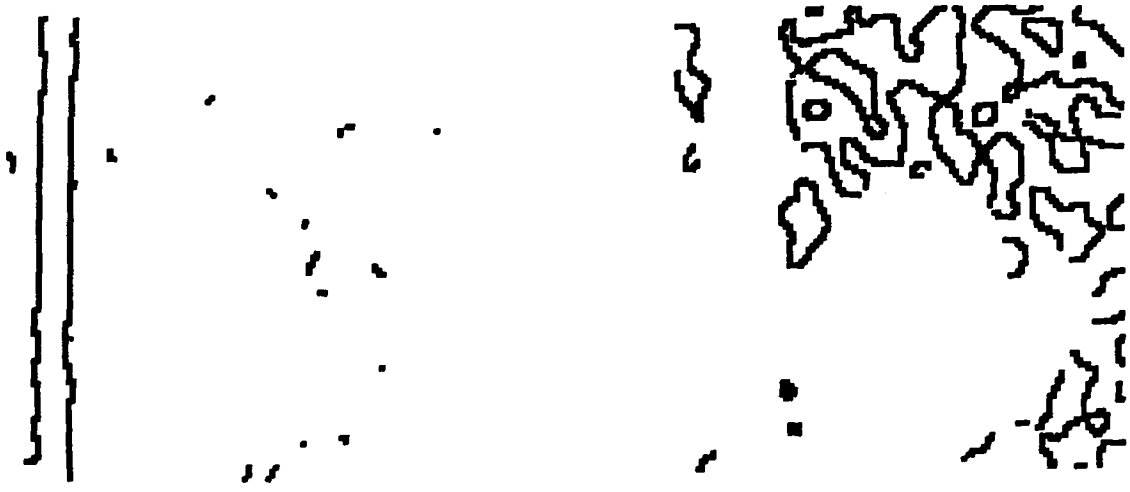


(j)

Figure 17, continued.

(g) Sampling over hypothesized FOEs represented with respect to a polar coordinate system which is not directly registered with the image coordinate system. Points of low intensity correspond to low global error. (h) Contour plot of (g) -- note marked minimum. (i) Displacements computed from the FOE corresponding to the minimum error FOE determined in (h). (j) Shows the depth histogram computed from the displacements in (i). The horizontal (x-) axis are units representing time until contact (i.e., units of camera displacement along the z-axis).

(k)



(l) (m)

Figure 17, continued.

(k,l,m) Extracted contour points corresponding to the three clusters in the histogram; (k) sign; (l) telephone pole; (m) trees.

reduced over the usual two-dimensional correlation process. In cases of the incorrect FOE there is a strong probability that many points will have poor correlations at all positions along the hypothesized displacement path. The shape of the resulting error function can be improved by selection of "interesting" image points of high contrast (boundaries) and high curvature (corners).

The determination of the translational motion parameters has now become a search process using a global error measure which is the sum of the errors of the best match on each point's flow path. The search process consists of two phases: a global sampling of the error measure, and then a local search at a finer sampling to determine the minimum. The error function appears to be very well behaved in a series of experiments on real scenes, and the algorithm seems rather robust.

In the case of pure rotation, the basic technique can be applied with minor differences. The search space for the correct rotational parameters is three-dimensional: two parameters for the axis of rotation and one for the magnitude of rotation. The algorithm can proceed in the same manner by choosing a set of distinguished points, and then compute a global error on a coarsely sampled parameter space. This problem is actually slightly more constrained than the first, because here the third dimension (amount of rotation) will directly constrain the image motion of all points simultaneously, while in the translational case each point had to be matched independently (because of differences in environmental depth).

In the case of motion restricted to a known plane, there are only two degrees of freedom. Translational motion will be constrained to the one dimension of the line represented by the intersection of the known plane and the image plane. The axis of rotation must be perpendicular to the plane, and therefore we must only determine the degree of rotation.

A set of experiments have proven these algorithms to be very robust in real scenes, including the outdoor road sequence from William's thesis [WIL81], industrial image domains supplied by the General Electric Corporation, and image sequences obtained in our laboratory.

V.3. Recovery of Depth via Occlusion Boundaries during General Sensor Motion

As we have pointed out earlier, the flow fields produced by a sensor undergoing general motion are difficult to interpret until they have been decomposed into their rotational and translational components. Once this has taken place, environmental depth can be recovered from translational displacements. Analytical techniques for performing this computation are extremely complex and can be quite sensitive to the errors that are typical in the computation of displacement fields. It is not feasible to exploit the approach of the previous cases where potential motion parameters were tested by computing a global error measure of lack of consistency across a set of image features. In the previous cases the dimensionality of the search space was no greater than three, but here it is a five-dimensional search space, and the computational demands may be excessive. In addition the error function cannot be expected to be well-behaved so that simple optimization techniques probably would not work.

Recently Lawton and Rieger [LAW83b] have described a surprisingly simple technique that promises to be rather robust in noisy, low resolution and/or sparse displacement fields. It depends upon the scene containing a sufficient number of significant depth discontinuities. Thus, a scene with several objects at distinct depths, or a single object of reasonable size against a textured background, will permit this technique to be effective.

Consider distinct surface features at different depths on an occlusion boundary. The points will be very close to each other in the image, and therefore sensor rotation causes an approximately equal rotational displacement. Thus, the only significant difference in their image displacement is caused by a difference in translational displacement. This leads to an algorithm which will exploit nearby image points which are at different depths. Note, however, that occlusion need not be determined because rotational components can be removed by taking differences between all nearby flow vectors. The resulting difference vectors will represent differences in depth, and vectors of significant magnitude will represent differences in depth of occlusion boundaries. They will be oriented on radial flow lines, emanating from the instantaneous axis of translation which can be determined by an optimization procedure.

There are several approaches to determining the axis of translation, such as the use of a Hough transform to select the point that most nearly lies at the intersection of the difference vectors. Due to practical noise considerations, a global error measure is used to evaluate each possible value for the direction of the translational axis in a coarse to fine search. The error measure used is the sum of the magnitudes of the error angles of the difference vector field and the set of radial field lines. Once the instantaneous axis of translation is determined, then the rotational component is overconstrained, can be determined and then subtracted out. Environmental depth of image points can then be

computed from the translational displacement.

The algorithm is not quite so straightforward because there may not be many reliable image displacement vectors that are at different depths and near each other. To the degree that they are not at sufficiently different depths, their difference vector will be short and prone to error. To the degree that they are not near each other, their rotational components will differ and introduce error. Thus, practical considerations in the application of the algorithm remain. However, several experiments have shown very promising results.

It should be noted that occlusion boundaries of independently moving objects will not satisfy the conditions for applying this algorithm, and thus the next algorithm complements this work.

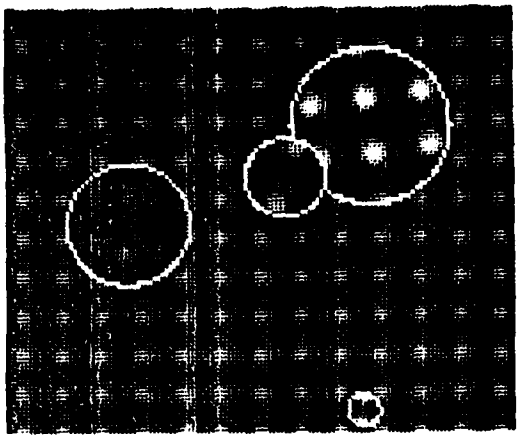
V.4. Scenes with Multiple Independently Moving Objects

The algorithms that we have just described do not confront the additional complexity introduced when there are multiple independently moving objects. The global types of constraints that were described earlier no longer apply across the entire image. The case of a sensor moving through a static environment can be equivalently viewed as an image of a single rigid object with associated motion parameters. However, if there are independently moving objects, they will have different motion constraints and introduce possibly serious errors in the global search of the parameter space for a single set of motion parameters. Thus, the goal is to decompose the image, and thereby separate the information in each flow field, so that motion of each object can be recovered.

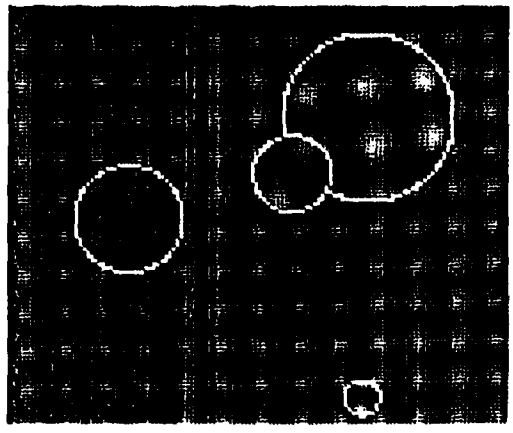
The approach outlined here is presented by Adiv [ADI83]. It involves a generalized Hough transform, proposing solutions to some of the problems found in this technique. Hough techniques are relatively insensitive to noise and can deal with partially incorrect or occluded data. Here, such a transform will be used to group a set of displacement vectors which satisfy the same motion parameters. However, there are a set of problems that must be considered: non-adjacent elements can vote for the same image transformation, there are difficulties in the detection of the motion parameters of small objects, and fine resolution of the motion parameter space can require large amounts of memory and computation time.

The suggested solution to these problems involves a modified multipass approach. In each pass windows are located around potential objects by the degree to which the displacement field is locally inconsistent with previously found motion transformations. The Hough transform is applied separately to the displacement vectors in each window. Thus, the sensitivity of the Hough transform to local events is increased and the motion parameters of small objects can be detected even in a noisy displacement field. A multiresolution scheme in both the image plane and the parameter space reduce the computational cost, while still maintaining accuracy.

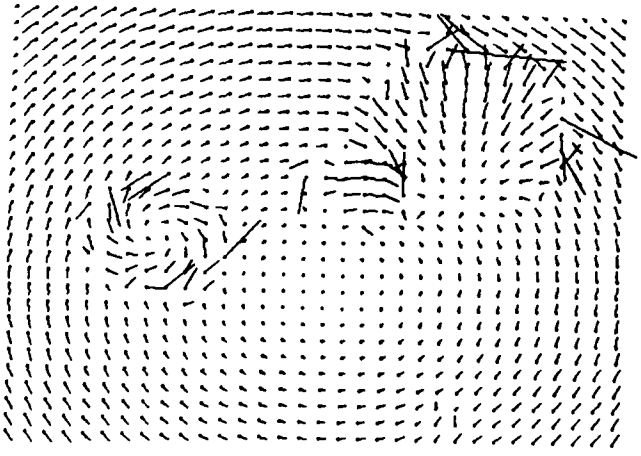
The algorithm has been shown to be efficient and robust in extracting motion parameters from artificial images with objects undergoing 2D motion as shown in Figure 18. It involves a 4-dimensional parameter space of horizontal translation, vertical translation, rotation (in the image plane) and expansion/contraction.



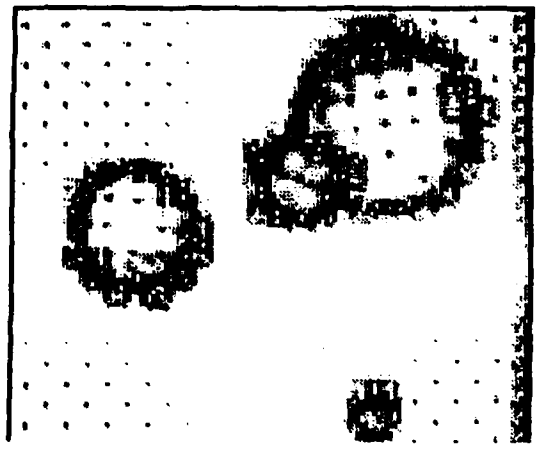
(a)



(b)



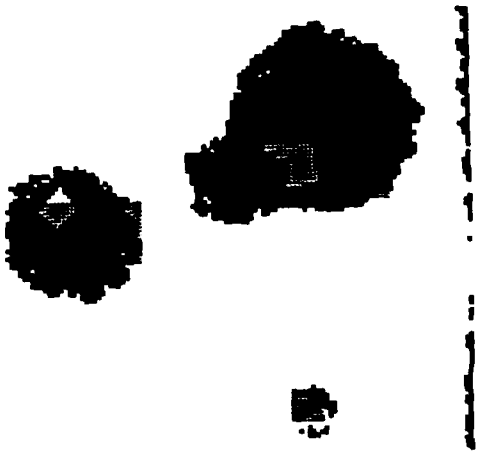
(c)



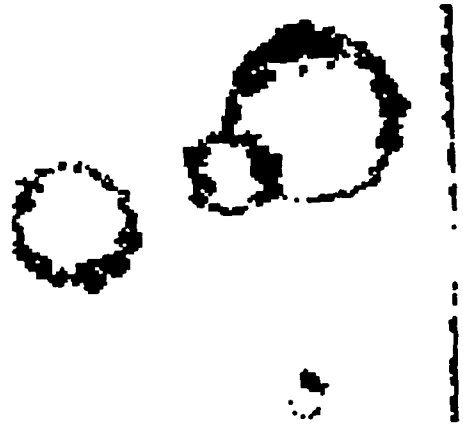
(d)

Figure 18. Determining Motion Parameters from Scenes Containing Multiple Independently Moving Objects.

(a,b) Intensity images; the white lines are included only to emphasize the objects and are not part of the images. Object A is the background, B is the circle in upper right corner, C is the circle partially occluding B, D is the circle in left part of image and E is the small circle in lower part. (c) Sampled displacement fields computed using the Horn/ Schunck method [HOR80]. (d) Weight plane computed from the displacement field. The dark areas represent incorrect values in the displacement field as measured by the degree to which the displacement vectors fail to satisfy smoothness constraints. These values are used to weight the 'vote' of each displacement vector in the Hough transform.



(e)



(f)

	rotation (radians)	expansion	vertical translation (pixels)	horizontal translation (pixels)
object A				
actual	0.025	0.	0.	0.
computed	0.0234	0.	0.	0.
object B				
actual	0.	0.1	−1.5	0.
computed	0.	0.09375	−1.2	0.
object C				
actual	−0.1	0.	0.	2.2
computed	−0.09375	0.	−0.2	2.1
object D				
actual	0.12	−0.1	0.	0.
computed	0.125	−0.0937	−0.2	0.
object E				
actual	0.	0.125	−1.1	0.7
computed	0.	0.0625 (*)	−1.05	0.6

(*) The large error indicated in this entry is due to the small size of object E (radius = 8 pixels) which reduces the possible resolution in the measurements of rotation and expansion.

(g)

Figure 18, continued.

(e) Optimal windows for which the corresponding displacement vectors are consistent with a computed motion transformation. The windows are determined using a multipass Hough technique which hypothesizes motion transformations by allowing each displacement vector to "vote" for potential transformations.
(f) Final results: the black areas correspond to incorrect values of displacement vector in the boundaries of objects. Each object now has a motion transformation associated with it.
(f) Comparison of actual and computed results.

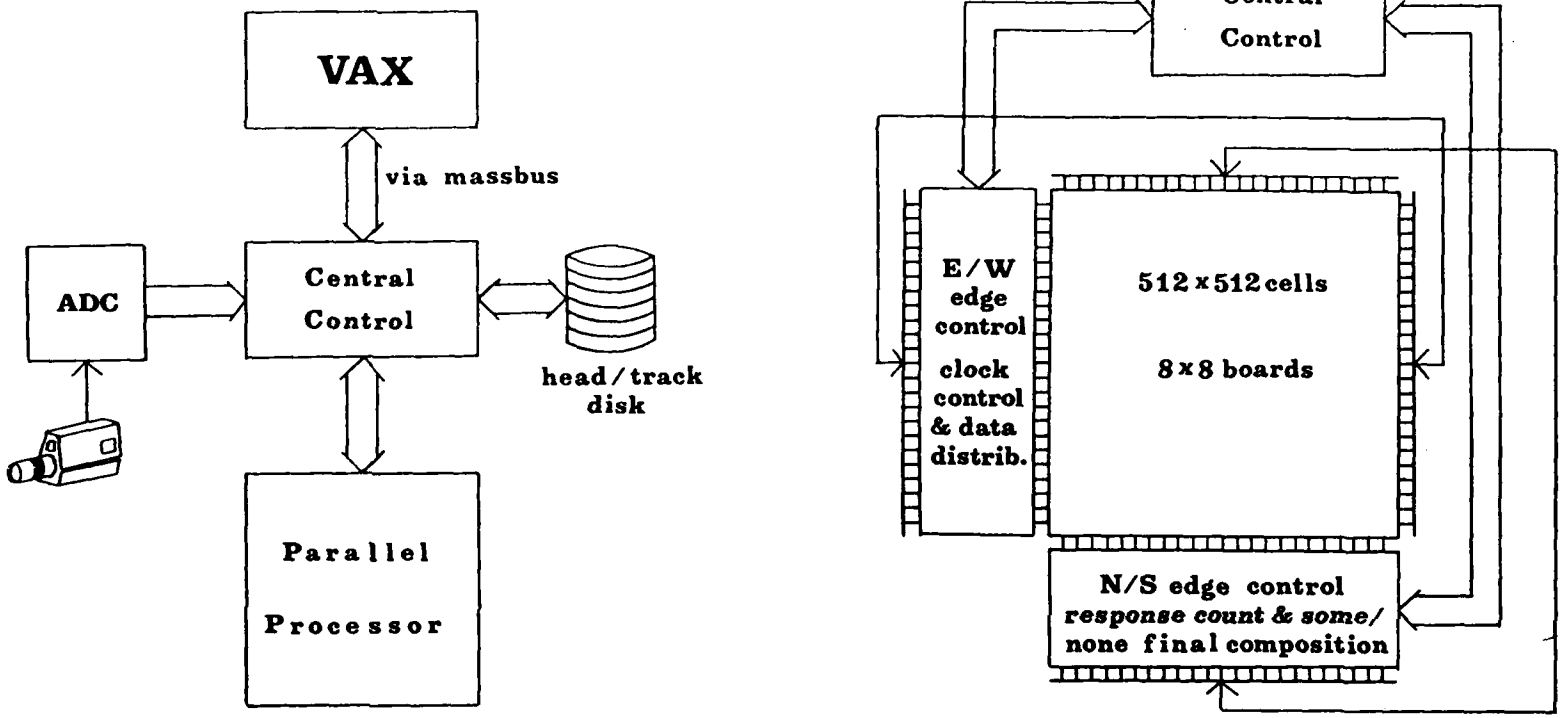
The current research involves the extension of this approach to 3D motion and to real scenes. This extension is non-trivial because displacement vectors in the 2D motion case involve four parameters with two constraints; thus, each displacement vector "votes" for a two-dimensional hyperplane of the parameter space. In the case of 3D motion when surface depth is unknown, there will be 5 motion parameters, and each displacement vector provides only one constraint; i.e., each will vote for a four-dimensional subspace of parameter cells. Thus, the signal to noise ratio in the parameter space will be much lower, and with the presence of noise in real images, the determination of peaks in generalized Hough space will be challenging.

VI. The CAAPP - A Highly Parallel Associative Architecture

Our research environment has maintained a continuous interest in parallel architectures and parallel algorithms. We estimate that real-time motion processing will require between one and two orders of magnitude more computational power than static vision. Thus, VLSI technology and massively parallel machines are obvious research directions.

Weems, Levitan, and Foster [WEE82] have developed a design for a Content Addressable Array Parallel Processor (CAAPP) and have been reformulating the motion algorithms with Lawton [WEE83a] for execution in this machine (Figure 19). The CAAPP is both a 512x512 Single Instruction Multiple Data (SIMD) array processor and an associative memory. The design is based on a 64x64 array of custom VLSI chips; it is intended to act as a slave processor for a general purpose computer system. Each chip contains 64 cells, an instruction decoder, and some miscellaneous logic. There are eight basic instruction types recognized by the chip, each performed in parallel by the constituent cells. Most instructions take one minor cycle time (100 nanoseconds) to execute. Inter-cell communication is bit serial and is accomplished by a four-way (N, S, E, W) cell interconnect network, allowing for three types of edge treatments: dead-edging, circular wrap, and zig-zag wrap. The entire memory may be bulk-loaded in one video frame time (1/30 second).

A very interesting application developed for the CAAPP (that makes use of the associativity and array processing capabilities) is an effective means of quickly and accurately decomposing a flow field into its rotational and translational components to recover the parameters of sensor motion. An exhaustive search procedure which implements top-down parallel correlation is used to determine which of a predefined set of rotational and translational

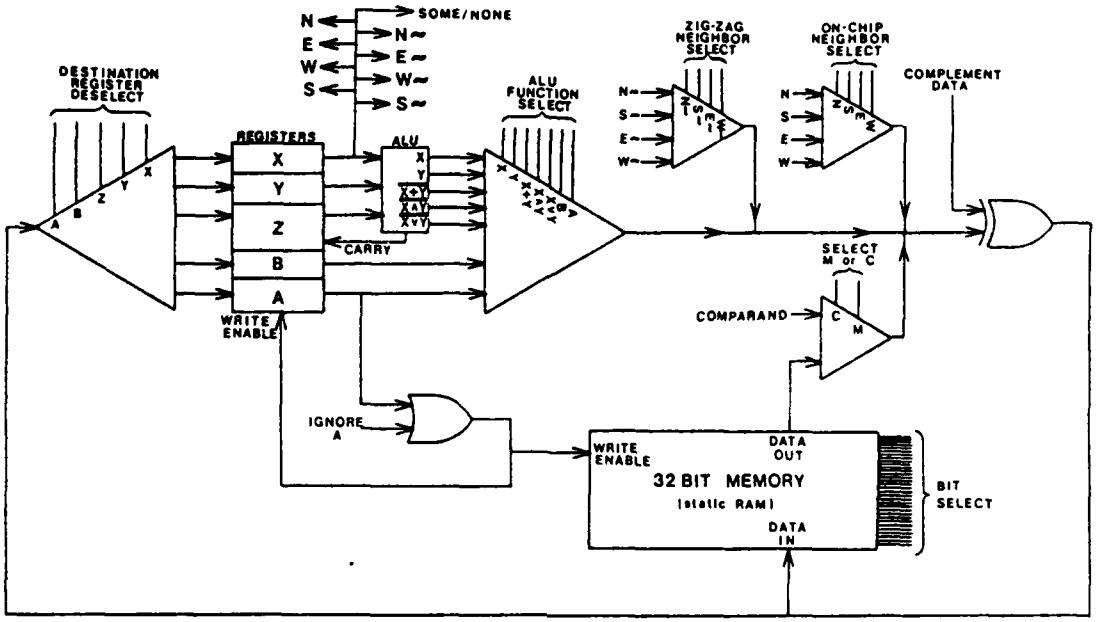


(a) (b)

Figure 19. Organization of CAAPP.

(a) The CAAPP is designed to function as a peripheral processor to an existing system; the local controller is responsible for assembling high level instructions from the host into sequences of CAAPP instructions and then executing those instructions, some of which require high speed data movement.

(b) The CAAPP is both a 512×512 SIMD array processor and an associative memory; the design is based on a 64×64 array of custom VLSI chips, each containing an 8×8 array of cells, an instruction decoder, and response logic.



Organization of One PE

(c)

Figure 19, continued.

(c) Each processing element recognizes eight basic instructions, most requiring 100 nanoseconds to execute. Inter-cell communication is bit-serial over a four-way interconnect network. Memory size and the need for 8-to-1 communication multiplexing is dependent on the implementation technology.

motion templates best account for the motion in a given flow field.

A set of 1000 rotational templates and 200 templates were obtained by uniformly sampling the motion parameters space and computing the 16x16 template corresponding to the sampled motion. The algorithm consists of four basic steps:

1. The rotational templates are loaded into the CAAPP.

2. A copy of the given flow field is loaded on top of each template location.

3. A difference field is formed by subtracting each rotational template from the flow field stored with it.

4. The similarity between the difference fields and each of the translational templates is evaluated, proceeding sequentially through the set of translational templates.

The flow field decomposition which is considered to be best is the rotational-translation pair which maximizes the similarity. This process is illustrated in Figure 20. Figure 20(a) is a representative flow field which is to be decomposed into a rotational-translational pair; Figure 20(b) is the same flow field shown in (a) but with random spike noise added. Figures 20(c) and 20(d) show the CAAPP response to the translational template which is closest to the actual translational motion of 20(a) and 20(b) respectively.

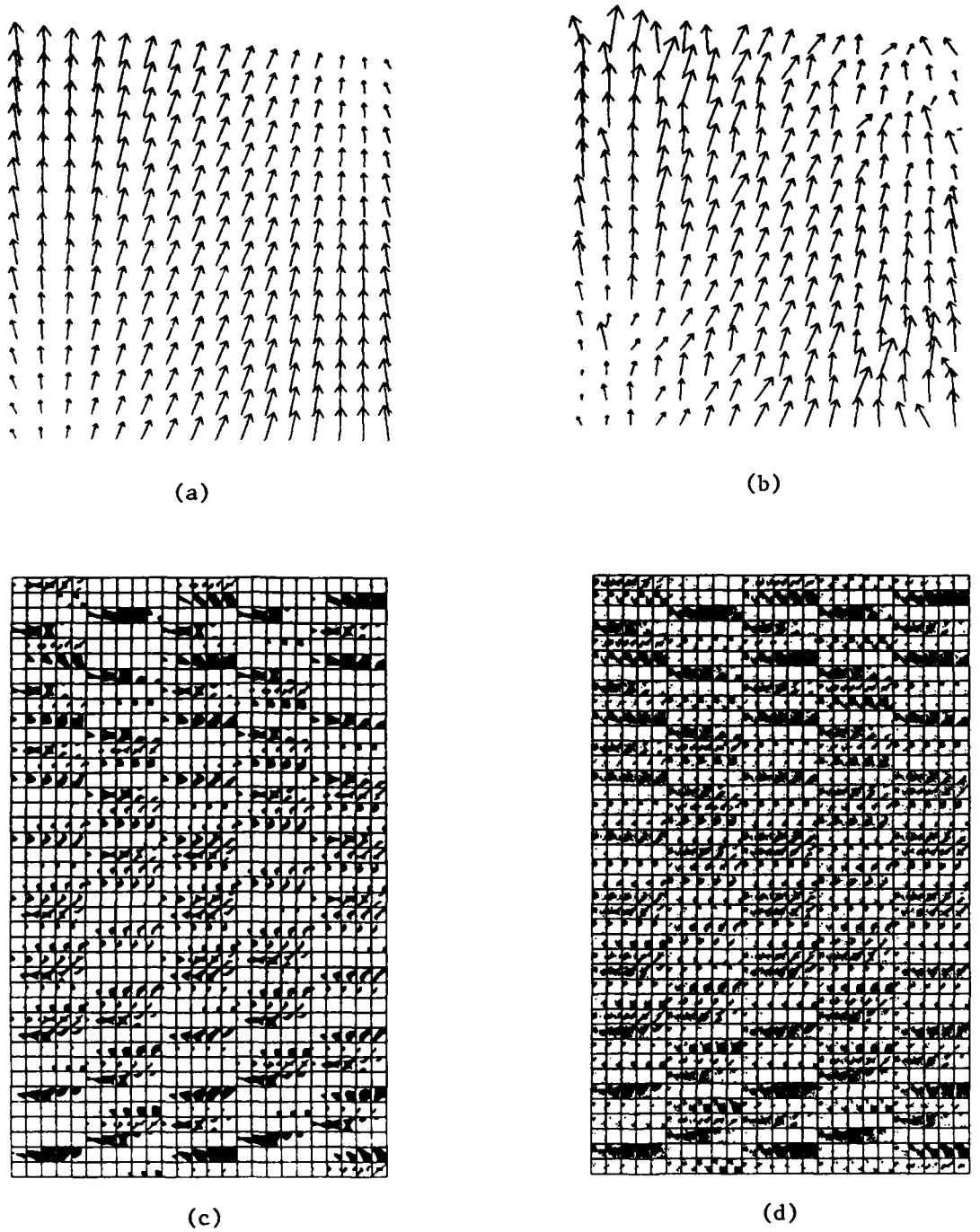
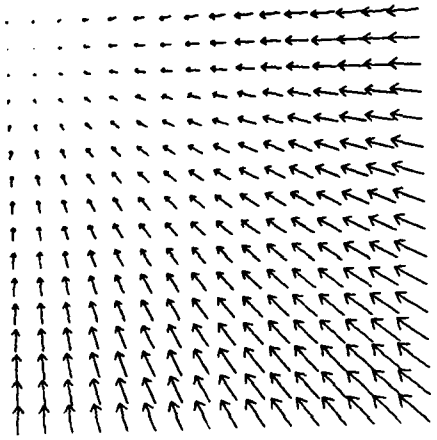
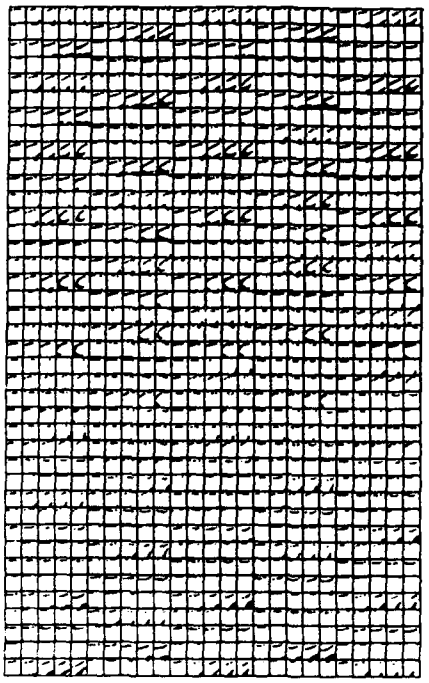


Figure 20. Motion Flow Field Decomposition in the CAAPP.

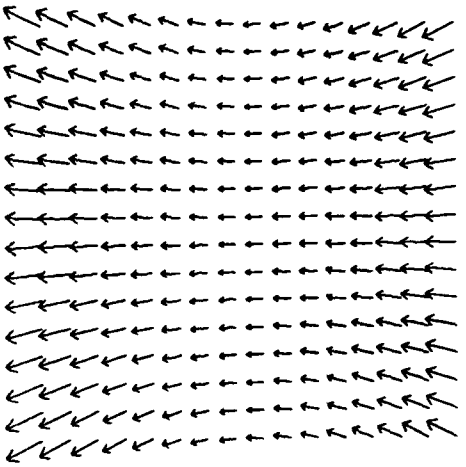
(a) A sample flow field with both rotational and translational components. (b) Flow field with random noise added. (c,d) CAAPP response for the translational template closest to the actual translational motion in the input field in (a) and (b), respectively.

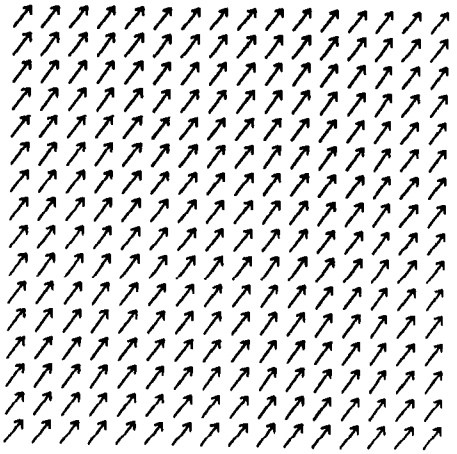


(e)



(f)





(g)

Figure 20, continued.

(e) Translational template which represents a motion not close to the correct translational motion. (f) Corresponding CAAPP response. (g) Rotational (upper) and translational (lower) templates selected by the algorithm.

Each 16x16 square represents a template position; a black dot within a square represents a position in a difference field where the similarity between the corresponding difference vector and translational vector exceeded a threshold. Thus, a perfect match of all 256 flow vectors (within the acceptable threshold) would produce a purely black square. For comparison purposes, Figure 20(c) shows a translational template which is not close to the actual translational motion in the fields in 20(a,b) and Figure 20(f) shows the corresponding CAAPP response. The rotational-translational pair chosen by the algorithm are shown in Figure 20(g); thse match the original fields very well. Note that there are a set of cells which respond fairly strongly, but are not spatially contiguous in the CAAPP. In actuality these cells all cluster near each other in the rotational-translational parameter space, but are physically separated in the diagram due to the two dimensional structure of the CAAPP and the way in which the fields were loaded.

Experiments have been performed with a CAAPP simulator on a VAX 11/780 using a wide variety of motions and simulated environments. In all cases examined, the translational template closest to the actual translational motion was selected. The rotational template was always close to the actual rotational motion, but was sometimes not the closest template. The procedure proved to be resistant to limited Gaussian noise as well as to limited random spike noise in the original flow field. The CAAPP timing calculations revealed that the algorithm could perform the rotational-translational decomposition in slightly more than 1/4 second. Given fabrication techniques available in the immediate future, execution times can be expected to be significantly improved.

Using the CAAPP strictly as a parallel array processor it is of course possible to perform standard image processing operations such as convolution. For example, a simple 3x3 Gaussian mask convolution can be done in 98 microseconds on the CAAPP. It should be noted that the time required to perform a convolution on the CAAPP is constant for a given image size and only varies depending on the size and complexity of the mask. A 10x10 mask of 8 bit multipliers applied to an image of 16 bit pixels (with the same number of pixels as the previous example) would require on average approximately 30 milliseconds (about one frame time). The method used is not restricted to square masks and is actually easily adapted to such shapes as annuli and disjoint areas.

VII. The Laboratory for Computer Vision Research

VII.1. Personnel

Director: Prof. Edward M. Riseman

Associate Director: Prof. Allen R. Hanson

Computer Lab Manager: Mr. Joey Griffith

Research Associates: Dr. Les Kitchen (Ph.D., 1982, University of Maryland)
Dr. Daryl Lawton (Ph.D., February 1984, University of Massachusetts)
Dr. George Reynolds (Ph.D., 1974, Wesleyan University)
Dr. Rich Weiss (Ph.D., 1976, Harvard University)

Visitors: Mr. Hiromasa Nakatani, Shizuoka University (7/83 - 9/84)
Prof. G. Burt Shaw, University of Oregon (Summer 1982, Summer 1983)

Ph.D. Theses:

John Prager (5/79): Segmentation of Static and Dynamic Scenes.

Paul Nagin (7/79): Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation.

Thomas Williams (5/81): Computer Interpretation of Dynamic Images from a Vehicle in Motion.

Bryant York (5/81): Shape Representation in Computer Vision.

John Lowrance (9/82): Dependency-Graph Models of Evidential Support.

Ralf Kohler (9/83): Integrated Non-Semantic Knowledge for Image Segmentation.

Daryl Lawton (2/84): Processing Dynamic Image Sequences from a Moving Sensor.

Steve Levitan* (5/84): Parallel Algorithms and Architectures: A Programmers Perspective.

Ph.D. Candidates:

G. Adiv, "The Interpretation of Optical Flow Fields"
P. Anandan
B. Burns
F. Glazer, "Hierarchical Motion Analysis in Machine Vision"
C. Kohl
D. Strahman
L. Wesley, "A Possibilistic-Based Model for High-Level Computer Vision"
T. Weymouth, "Using Object Descriptions in a Schema Network for Machine Vision"
C. Weems*, "Image Processing on a Content Addressable Array Parallel Processor"

* with Parallel Architecture Group

M.S. Candidates	Programmers	Undergraduate Assistants
M. Boldt	R. Heller	R. Belknap
N. Irwin	D. Thompson	V. Cohen
J. Rieger		D. Ritscher
		K. Ward

VII.2. Funding: Recent Grants and Contracts (excluding equipment grants)

Air Force Office of Scientific Research, 4/83 - 3/85
Representation and Control in the Interpretation of Complex Scenes

Defense Advanced Research Projects Agency, 6/82 - 5/84
Processing Dynamic Images from Camera Motion

Office of Naval Research, 1/74 - 3/84
Semantically Directed Vision Processing

National Science Foundation, 9/79 - 8/82
A Computer System for Visual Interpretation of Natural Scenes

UMass Remote Sensing, 2/82 - 2/83
Development of Initial Design and Implementation Specifications for the University of Massachusetts Segment of Remote Sensing Research and Development Programs

Rome Air Development Center via Syracuse University, 8/83 - 1/84
Applying the VISIONS System to Interpretation of Aerial Images

Digital Equipment Corporation, 7/82 - 6/84
Image Analysis Applied to Industrial Automation

Tufts New England Medical Center, 7/81 - 6/82
Biomedical Image Analysis Applied to Ophthalmology

UMass Biomedical Research Support Grant, 4/81 - 3/83
Biomedical Image Analysis Applied to the Prognosis of Malignant Melanoma

General Electric, 4/83 - 12/83
Feasibility Study for the Construction of a Content Addressable Array Processor

A.C. Nielsen, 3/83 - 3/84
Image Processing

VIII. References

[ADI83] Adiv, G., "Recovering 2-D Motion Parameters in Scenes Containing Multiple Moving Objects," COINS Technical Report 83-11, University of Massachusetts at Amherst, May 1983.

[BUR83] Burns, J., Hanson, A. and Riseman, E., "Extraction of Linear Features," in preparation.

[DEM67] Dempster, A., "Upper and Lower Probabilities Induced by a Multivalued Mapping," Annals of Mathematical Statistics, 38, 1967, pp. 325-339.

[DEM68] Dempster, A., "A Generalization of Bayesian Inference," Journal of the Royal Statistical Society, Series B, 30, 1968, pp. 205-247.

[GLA81] Glazer, F., "Computing Optic Flow," Proc. IJCAI-7, 1981, pp. 644-677.

[GLA82] Glazer, F., "Multilevel Relaxation in Low Level Computer Vision," COINS Technical Report 82-30, University of Massachusetts at Amherst, 1982.

[GLA83a] Glazer, F., Reynolds, G. and Anandan, P., "Scene Matching by Hierarchical Correlation," Proc. of Computer Vision and Pattern Recognition Conference, Arlington, Virginia, June 1983.

[GLA83b] Glazer, F., "Multilevel Relaxation in Low Level Computer Vision," COINS Technical Report 82-30, University of Massachusetts at Amherst; also to appear in Multiresolution Image Processing and Analysis (A. Rosenfeld, Ed.), Springer-Verlag, 1983.

[HAN74] Hanson, A. and Riseman, E., "Preprocessing Cones: A Computational Structure for Scene Analysis," COINS Technical Report 74C-7, University of Massachusetts at Amherst, September 1974.

[HAN75] Hanson, A. and Riseman, E., "The Design of a Semantically Directed Vision Processor (Revised and Updated)," COINS Technical Report 75C-1, University of Massachusetts at Amherst, February 1975.

[HAN78a] Hanson, A. and Riseman, E., "Segmentation of Natural Scenes," in Computer Vision Systems (A. Hanson and E. Riseman, eds.), Academic Press, 1978, pp. 129-163.

[HAN78b] Hanson, A. and Riseman, E., "VISIONS: A Computer System for Interpreting Scenes," in Computer Vision Systems (A. Hanson and E. Riseman, eds.), Academic Press, 1978, pp. 303-333.

[HAN80a] Hanson, A., Riseman, E. and Glazer, F., "Edge Relaxation and Boundary Continuity," COINS Technical Repirt 80-11, University of Massachusetts, May 1980.

[HAN80b] Hanson, A. and Riseman, E., "Processing Cones: A Computational Structure for Image Analysis," in Structured Computer Vision (S. Tanimoto, Ed.), Academic Press, New York, 1980. (Also COINS Technical Report 81-38, December 1981).

[HOR80] Horn, B.K.P. and Schunck, B.G., "Determining Optical Flow," MIT A.I. Memo 572, 1980.

[KOH81] Kohler, R., "A Segmentation System Based on Thresholding," Computer Graphics and Image Processing, 15, 1981, pp. 319-338.

[KOH82] Kohler, R. and Hanson, A., "The VISIONS Image Operating System," Proc. of 6th International Conference on Pattern Recognition, Munich, Germany, October 1982.

[KOH83] Kohler, R., "Integrated Non-Semantic Knowledge for Image Segmentation," Ph.D. Thesis, Computer and Information Science Department, University of Massachusetts at Amherst, September 1983.

[LAW82] Lawton, D., "Motion Analysis via Local Translational Processing," IEEE Workshop on Computer Vision: Representation and Control, Rindge, New Hampshire, August 1982.

[LAW83a] Lawton, D., "Processing Restricted Motion," Proc. of the DARPA Image Understanding Workshop, Arlington, Virginia, June 1983.

[LAW83b] Lawton, D. and Rieger, J., "The Use of Difference Fields in Processing Sensor Motion," Proc. of the DARPA Image Understanding Workshop, Arlington, Virginia, June 1983.

[LAW83c] Lawton, D., "Processing Translational Motion Sequences," Computer Vision Graphics and Image Processing, April 1983.

[LAW84] Lawton, D., "Processing Dynamic Image Sequences from a Moving Sensor," Ph.D. Thesis and COINS Technical Report 84-05, University of Massachusetts at Amherst, February 1984.

[LOW78] Lowrance, J., "GRASPER 1.0 Reference Manual," COINS Technical Report 78-20, University of Massachusetts at Amherst, December 1978.

[LOW79] Lowrance, J. and Corkill, D., "The Design of GRASPER 1.0: A Graph Processing Programming Language Extension," COINS Technical Report 79-6, University of Massachusetts at Amherst, February 1979.

[McC82] McCormick, C., "Strategies for Knowledge-Based Image Interpretation," COINS Technical Report 82-10, University of Massachusetts at Amherst, May 1982.

[NAG79] Nagin, P., "Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation," Ph.D. Thesis, COINS Technical Report 79-15, University of Massachusetts at Amherst, September 1979.

[NAG81a] Nagin, P., Hanson, A. and Riseman, E., "Region Relaxation in a Parallel Hierarchical Architecture," in Real-Time Parallel Computing Image Analysis (M. Onoe, K. Preston and A. Rosenfeld, Eds.), Plenum Publishing Corporation, 1981.

[NAG81b] Nagin, P., Hanson, A. and Riseman, E., "Variations in Relaxation Labelling Techniques," Computer Graphics and Image Processing, 17, 1981, pp. 33-51.

[OVE79] Overton, K. and Weymouth, T., "A Noise Reducing Preprocessing Algorithm," 1979 IEEE Pattern Recognition and Image Processing Conference, Chicago, August 1979.

[PAR80] Parma, C., Hanson, A. and Riseman, E., "Experiments in Schema-Driven Interpretation of a Natural Scene," COINS Technical Report 80-10, University of Massachusetts, 1980. Also in Nato Advanced Study Institute on Digital Image Processing (R. Haralick and J.C. Simon, eds.), Bonas, France, 1980.

[PRA79] Prager, J., "Segmentation of Static and Dynamic Scenes," Ph.D. Thesis, COINS Technical Report 79-7, University of Massachusetts at Amherst, May 1979.

[PRA80] Prager, J., "Extracting and Labeling Boundary Segments in Natural Scenes," IEEE Trans. Pattern Analysis and Machine Intelligence, Volume PAMI-2, January 1980, pp. 16-27.

[PRA83] Prager, J. and Arbib, M., "Computing the Optic Flow: The MATCH Algorithm and Prediction," Computer Vision, Graphics, and Image Processing 24, pp. 271-304, April 1983.

[REY84a] Reynolds, G., Kitchen, L., Riseman, E. and Hanson, A., "Experiments in Automatic Feature Extraction - A Report to RADC," February 1984.

[REY84b] Reynolds, G., Irwin, N., Hanson, A. and Riseman, E., "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," Proc. of the 1984 Computer Vision Conference, Annapolis, MD, April 30 - May 2, 1984.

[RIE83] Rieger, J., "Information in Optical Flows Induced by Curved Paths of Observation," Journal of Optical Society of America, Volume 73, to appear, 1983.

[SHA76] Shafer, G., A Mathematical Theory of Evidence, Princeton University Press, 1976.

[SHA83a] Shaw, G., "Determining Motion Parameters Using a Perturbation Approach," COINS Technical Report 83-30, University of Massachusetts at Amherst, September 1983.

[SHA83b] Shaw, G., "Some Remarks on the Use of Color in Machine Vision," COINS Technical Report 83-31, University of Massachusetts at Amherst, September 1983.

[WEE82] Weems, C., Levitan, S. and Foster, C., "Titanic: A VLSI-Based Content Addressable Parallel Array Processor," Proc. of IEEE International Conference on Circuits and Computers, September 1982, pp. 236-239.

[WEE83a] Weems, C., Levitan, S., Lawton, D. and Foster, C., "A Content Addressable Array Parallel Processor and Some Applications," Proc. of the DARPA Image Understanding Workshop, Arlington, Virginia, June 1983.

[WEE83b] Weems, C., Levitan, S. and Foster, C., "Titanic: A Content Addressable Array Parallel Processor," COINS Technical Report 83-32, University of Massachusetts at Amherst, October 1983.

[WES82] Wesley, L. and Hanson, A., "The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the VISIONS System," in Proc. of the Workshop on Computer Vision, Rindge, New Hampshire, August 1982, pp. 14-25.

[WES83] Wesley, L., "Reasoning About Control: An Evidential Approach," SRI Tech. Memo, to appear.

[WEY83] Weymouth, T., Griffith, J., Hanson, A. and Riseman, E., "Rule Based Strategies for Image Interpretation," Proc. of the DARPA Image Understanding Workshop, Arlington, Virginia, June 1983. A shortened version also appeared in Proc. AAAI, Washington, D.C., August 1983.

[WEY84] Weymouth, T., "Using Object Descriptions in a Schema Network for Machine Vision," Ph.D. Thesis, COINS Department, University of Massachusetts at Amherst, in progress.

[WIL77] Williams, T. and Lowrance, J., "Model-Building in the VISIONS High Level System," COINS Technical Report 77-1, University of Massachusetts at Amherst, January 1977.

[WIL80] Williams, T., "Depth from Camera Motion in a Real World Scene," IEEE PAMI, Volume PAMI-2, Number 6, November 1980, pp. 511-516.

[WIL81] Williams, T., "Computer Interpretation of a Dynamic Image from a Moving Vehicle," Ph.D. Thesis and COINS Technical Report 81-22, University of Massachusetts at Amherst, May 1981.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER COINS TR 83-35	2. GOVT ACCESSION NO. AD-A150 818	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A SUMMARY OF IMAGE UNDERSTANDING RESEARCH AT THE UNIVERSITY OF MASSACHUSETTS		5. TYPE OF REPORT & PERIOD COVERED INTERIM 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Allen R. Hanson Edward M. Riseman		8. CONTRACT OR GRANT NUMBER(s) ONR N00014-75-C-0459 DARPA N00014-82-K-0464
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer and Information Science Department University of Massachusetts Amherst, Massachusetts 01003		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE October 1983 13. NUMBER OF PAGES 91
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

segmentation
image interpretation
image pyramids
motion
knowledge representation
inferencing
associative array architectures
feature matching
image registration

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The major focus of our research program revolves around issues of static and dynamic image understanding. Our principle objective in this work is to confront fundamental problems in computer vision in the context of a large scale experimental system for interpretation of complex images. In this report we briefly review the current status of the VISIONS image understanding system, focussing on:

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

- the extraction of low-level syntactic descriptions of images,
- the representation of knowledge in a form suitable for use in the interpretation process,
- strategies for utilizing modular knowledge sources to link the sensory data to semantic hypotheses,
- inference mechanisms for integrating ambiguous and partial evidence from multiple sources, and
- control methodologies for both data-directed and knowledge-directed interpretation processes.

Our work in dynamic image interpretation (motion) is concerned with techniques for recovery of environmental information, such as depth maps of the visible surfaces, from a sequence of images produced by a sensor in motion. Algorithms that appear robust have been developed for constrained sensor motion such as pure translation, pure rotation, and motion constrained to a plane. Interesting algorithms with promising preliminary experimental results have also been developed for the case of general sensor motion in images where there are several significant depth discontinuities, and for scenes with multiple independently moving objects. A general hierarchical parallel algorithm for efficient feature matching has also been developed for applications in motion, stereo, and image registration. In addition, we have been designing a highly parallel architecture that integrates aspects of both parallel array processing and associative memories for real-time implementation of motion algorithms.

END

FILMED

4-85

DTIC